



Europäisch Patentamt

European Patent Offi

Offic européen de brevets



(11) Numér de publication : 0 574 303 A1

(12)

DEMANDE DE BREVET EUROPEEN

(21) Numéro de dépôt : 93401459.8

(51) Int. Cl.⁵ : G06F 9/46

(22) Date de dépôt : 08.06.93

(30) Priorité : 09.06.92 FR 9206933

(43) Date de publication de la demande : 15.12.93 Bulletin 93/50

(84) Etats contractants désignés : DE ES FR GB IT

(71) Demandeur : BULL S.A.
Tour BULL, 1, place Carpeaux
F-92800 Puteaux (FR)(72) Inventeur : Li, Jianzhong
7, le mail des Houssières
F-92290 Chatenay-Malabry (FR)
Inventeur : Abdelloumni, Mohammed
95, rue Duuhesme
F-75018 Paris (FR)
Inventeur : Brunon, Jean-Philippe
4, rue Darwin
F-75018 Paris (FR)(74) Mandataire : Debay, Yves et al
BULL S.A. Tour BULL Cédex 74 PC/TB2803
F-92039 Paris La Défense (FR)

(54) Système de traitement transactionnel entre un serveur informatique et une pluralité de stations de travail.

(57) La présente invention concerne un système de traitement transactionnel entre un serveur (3) informatique et une pluralité de stations de travail (2) entre lesquelles les ouvrages correspondants à des activités constituant des procédures sont exécutés, le système comportant du côté serveur un module ordonnanceur pour ordonner le progrès des ouvrages, un module répartiteur (31) pour effectuer la répartition des tâches qui s'appuient sur la relation entre rôle et acteur, un module signaleur (34) pour conserver une trace des événements rythmés et recevant des informations de l'ordonnanceur (30) et d'un module coordinateur (31), une boîte de messages (35) échangeant des signaux d'une part avec les modules répartiteur (33) et signaleur (34), et d'autre part avec un module de communication (36) avec les stations de travail et un module de gestion des données du flux de travail (370), le système comportant du côté station de travail une couche de communication (27), une couche de gestion des messages (26) correspondant au flux de travail, une couche (24) comportant différents jeux d'interfaces de programmation de l'application, une couche de services de gestion (25) et une couche d'interface utilisateur (20,21).

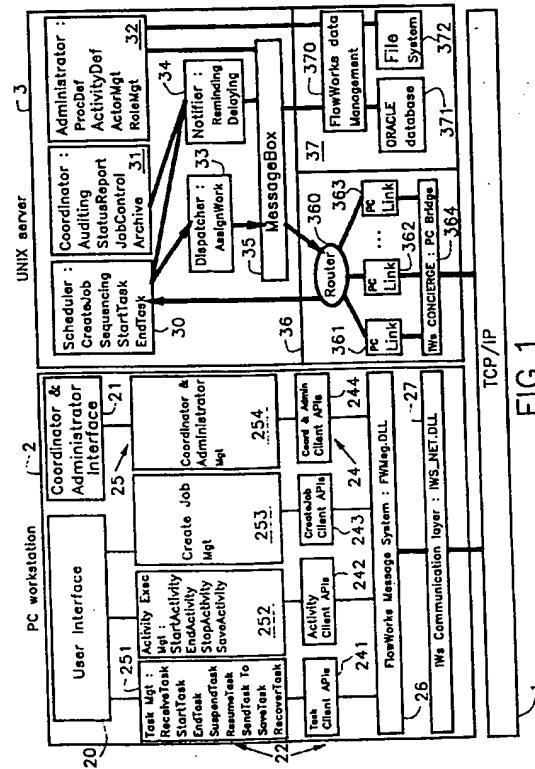


FIG. 1

EP 0 574 303 A1

Jouve, 18, rue Saint-Denis, 75001 PARIS

L'invention concerne un système de traitement transactionnel (Programme d'application coopérative) entre un serveur informatique et une pluralité de stations de travail.

Le but de la présent invention est de faire collaborer les agents (humains ou automatiques) d'une unité organisationnelle (par exemple un département) à travers des procédures (définies à l'aide du formalisme ICN - Information Control Net), et de suivre et contrôler le progrès des exécutions (matérialisé par des ouvrages) de procédures. Les procédures sont définies en concordance avec les objectifs de l'unité organisationnelle. Les procédures définissent le séquencement des étapes auxquelles un ou plusieurs agents doivent intervenir ou contribuer pour atteindre l'objectif visé par la procédure.

L'architecture proposée permet également d'élargir le champ d'application des systèmes multi-tâches à toute sorte d'applications sans préjuger au préalable de l'application, seul le découpage des procédures est effectué selon des règles définies liées à la forme de l'architecture.

Ce but de l'invention est obtenu par le fait que le système de traitement transactionnel entre un serveur informatique et une pluralité de stations de travail entre lesquelles les ouvrages correspondants à des activités constituant des procédures sont exécutés, le système comportant du côté serveur un module ordonnanceur des ouvrages qui s'appuient sur la définition des procédures (à savoir les précédences entre les activités d'une procédure) mémorisées dans la base de données relationnelle pour ordonner le progrès des ouvrages en trouvant la ou les activités suivantes et en créant la ou les tâches correspondantes étant donnée la fin d'une activité, un module répartiteur pour effectuer la répartition des tâches qui s'appuie sur la relation entre rôle et acteur mémorisée dans la base de données relationnelle pour attribuer une tâche à un acteur choisi éventuellement selon la charge de travail, un module signaleur pour conserver une trace des événements rythmés et recevant des informations de l'ordonnanceur et d'un module coordinateur, une boîte de messages échangeant des signaux d'une part avec les modules répartiteur et signaleur, et d'autre part avec un module de communication avec les stations de travail et un module de gestion des données du flux de travail, le système comportant du côté station de travail une couche de communication, une couche de gestion des messages correspondant au flux de travail, une couche comportant différents jeux d'interfaces de programmation de l'application, une couche de services de gestion et une couche d'interface utilisateur.

Selon une autre particularité les différents modules d'une couche de la station de travail n'ont pas d'interaction entre eux.

Selon une autre particularité le module de gestion des données gère des tables statiques et dynamiques mémorisées dans la base de données relationnelle.

Selon une autre particularité les tables statiques sont constituées des données de définition des procédures, de définition des activités, de l'organisation des acteurs et des rôles et de répartition des activités selon une table de distribution.

Selon une autre particularité les tables dynamiques sont constituées de données évolutives représentant l'avancement des ouvrages et des tâches.

Selon une autre particularité les tables dynamiques sont constituées d'une table de circulation, d'une table de vecteurs de données d'ouvrage JDV-JDV, d'une table de vecteurs de données d'activité ADV-ADV et des tables SCHED-STATEP, SCHED-STATUSP, SCHED-JOBS, SCHED-TASKP.

Selon une autre particularité, les tables statiques sont constituées d'une table de description de procédure (SCHED-PROCEDURE), d'une table de description de données de procédure (JDV-PDDEF), d'une table de description des références des données stockées dans des bases de données externes (EDA-ORA), d'une table de description des activités (SCHED-ACTIVITE), d'une table de définition des précédences (SCHED-PRECEDENCE), des tables de définition de l'organisation du système (ORGA-ACTOR, ORGA-ROLE, ORGA-R-OFF-ACT).

Selon une autre particularité, une activité est constituée d'un identifiant de l'activité, et de la procédure à laquelle l'activité est rattachée et d'une liste d'actions comportant une commande d'initialisation et une commande de validation.

Selon une autre particularité, les actions sont constituées d'une information formant le type, d'un nom, d'argument constitué par les données des vecteurs JDV ou ADV.

Selon une autre particularité la couche de service de gestion comporte :

- un service de gestion de création d'ouvrage,
- un service de gestion d'administration et de coordination,
- un service de gestion de l'exécution de l'activité.

Selon une autre particularité, une activité comporte également un préfixe PR destiné à appeler un logiciel de présentation et d'interaction fournissant l'interface de l'activité.

Selon une autre particularité, le module administrateur réalise les fonctions de modélisation, création, test et accord des différentes entités du système.

Selon une autre particularité, le module de gestion de données a pour fonction d'attacher les données ap-

propriétés aux ouvrages spécifiques en créant le vecteur JDV, de gérer l'intégrité des données, de retrouver et accéder les données dans les bases de données externes (ou d'application).

5 Selon une autre particularité, le module de gestion d'exécution d'activités comporte les fonctions:

- d'lecture du fichier et de description d'activité,
- de lecture et d'écriture des données d'ouvrage utilisées dans l'activité,
- de lecture et d'écriture des données des tâches correspondantes utilisées dans l'activité et de commandes des droits d'accès sur les données d'ouvrage,
- d'avancement du formulaire de présentation approprié de l'activité,
- et de l'exécution des actions correspondantes à l'activité d'une façon guidée.

10 Selon une autre particularité, le module de gestion des tâches réalise les fonctions suivantes :

- réception d'une tâche offerte,
- acceptation d'une tâche offerte,
- recherche des données correspondantes du vecteur d'ouvrage sur le serveur,

15 - achèvement d'une tâche,

- restitution du vecteur d'ouvrage et mise à jour au serveur,
- réorientation d'une tâche vers un autre acteur ou une autre activité,
- appel de l'exécution de l'activité d'une tâche,
- traitement et maintien local des données correspondant aux tâches et leur historique.

20 Selon une autre particularité le module administrateur et coordinateur (11) comporte les fonctions d'édition de concept et de données et la commande des ouvrages.

Selon une autre particularité, le module répartiteur réalise trois fonctions principales :

- affecter à un utilisateur les tâches qui viennent d'être créées par l'ordonnanceur,
- affecter un nouvel utilisateur à une tâche qui vient d'être rejetée,
- effectuer la distribution et la circulation des activités parmi les utilisateurs.

25 Selon une autre particularité, le module signaleur conserve une trace des événements rythmés ou temporelle et comporte une fonction de rappel quand un temps est expiré et une fonction de retardement pour lancer un travail avec un retard.

30 Selon une autre particularité, la boîte de messages est constituée de deux tables de données, une première qui contient la description des messages et une seconde qui mémorise la destination des messages.

D'autres particularités et avantages de la présente invention apparaîtront plus clairement à la lecture de la description ci-après faite en référence aux figures ci-après dans lesquelles :

La figure 1 représente de façon schématique un système de traitement transactionnel selon l'invention.

La figure 2 représente de façon schématique les échanges entre les différents modules du système de traitement transactionnel.

35 La figure 3 représente les informations qui transitent lors de la création d'un ouvrage entre les différents éléments du système transactionnel.

La figure 4 représente l'écran affiché par le terminal du système lors de la mise en route de celui-ci avec l'interface de présentation particulière.

40 Les figures 5 à 15 représentent les différentes fenêtres utilisées par l'interface utilisateur du système transactionnel.

La figure 1 représente un système de traitement transactionnel entre un serveur informatique (3) utilisant des données de traitement stockées sur une base de données relationnelle d'un serveur informatique (3), par exemple du type "UNIX", et communiquant avec des stations de travail (1), par exemple du type PC, sur lesquelles le travail d'une application qui se déroule dans le cadre d'activités est exécutée par l'utilisateur de la station de travail.

45 Le serveur (3) comporte un jeu de modules (30 à 37) qui initialisent et coordonnent les activités. Par activité on doit entendre le corps d'une étape de travail, d'une procédure, c'est à dire une unité de base de travail qui doit être un jeu séquentiel d'actions primitives exécutées par un seul acteur.

50 Les activités peuvent fonctionner selon un des trois modes suivants : automatique, manuel et mixte.

Le corps d'une activité constitue la partie d'exécution d'une activité. Ce corps comporte une liste d'actions à exécuter sur la station de travail. L'ordre de ces actions n'est pas fixé et dépend de l'interruption de l'acteur exécutant. Un corps d'activité comporte un identifiant et une liste d'actions Ai. Le couple [ProcedureId, ActivityId] de l'annexe 1 identifie uniquement un corps d'activité lié à l'activité (ActivityId) qui appartient à la procédure dont l'identité est (ProcedureId). Une action N peut être activée par l'utilisateur à travers le formulaire correspondant à l'écran de la station de travail (1). Cette action peut être scindée en deux parties : une première partie Ai qui est une commande (PreScript) à exécuter quand l'utilisateur veut sélectionner une action N (par exemple une initialisation) et une deuxième partie Zi qui est la commande (PostScript) à exécuter quand l'utilisateur veut valider l'action sélectionnée.

Les actions d'une activité sont numérotées de 0 à N. L'action AO est une action particulière dans laquelle la commande (PreScript) de sélection est automatiquement exécuté et validée avant toute autre action.

Le préfix PR de l'activité appelle un formulaire qui fournit l'interface de présentation et d'interaction de l'activité sur l'écran de la station.

La structure donnée de l'activité est fournie par un fichier dont le nom est précédé du préfixe ADV. La syntaxe d'une action est constituée d'un champ "type d'action" (ActionType) d'un champ "nom d'action" (ActionName) et de plusieurs arguments qui sont des champs de vecteurs de données d'ouvrage (JDV) représentés par JI et de vecteurs de données d'activité (ADV) représentés par AI, ces champs sont stockés dans les vecteurs (ADV, JDV) de la mémoire (370) du serveur (3). Les actions peuvent être du type : exécutable, fonction DLL, échanges DDE... Le système de traitement transactionnel aide à définir, exécuter, coordonner et commander le flux de travail entre les organisations ou les groupes de travail en se basant sur une description des tâches de bureau qui peuvent être décrites comme des tâches structurées récurrentes, appelées procédures dont les éléments de travail de base sont appelés activités et doivent être exécutées par différents individus appelés acteurs selon une certaine séquence.

Un acteur consiste en une identification unique de l'acteur (usrid) fournie par le système, un nom d'acteur (name), un prénom de l'acteur (firstName), un mot unique d'ouverture de session (loginName), un mot de passe (password) et une étiquette (alias).

Une procédure est un jeu prédéfini d'étapes de travaux selon un séquencement partiel de ces étapes. Une étape de travail est formée d'une en-tête (identification, précédence, etc...) et d'un corps (activité) qui constitue le travail actuel à exécuter. Différentes étapes d'une procédure peuvent être exécutées par différentes personnes, différents groupes de personnes. Dans certains cas, plusieurs étapes d'une procédure peuvent être exécutées en même temps ou dans un ordre quelconque. En général, une procédure est un jeu partiellement ordonné d'étapes, c'est à dire dans lequel toutes les étapes ne doivent pas être nécessairement exécutées séquentiellement et dans lequel des boucles sont autorisées, plutôt qu'un jeu totalement ordonné. Les procédures ont de façon classique des attributs tels que le nom et la personne responsable associée à la procédure.

Un script est une spécification d'une procédure, d'une activité ou d'une partie automatique d'une activité manuelle.

Un ouvrage (job) est le lieu de commande pour une exécution particulière d'une procédure. Dans certains contextes l'ouvrage est appelé une transaction. Les ouvrages ont des paramètres tels que l'état, l'initiateur, l'historique.

Un rôle (role) consiste en la désignation d'un acteur ou d'un groupe d'acteurs qui agissent comme base pour les commandes d'accès et les commandes d'exécution. Un acteur peut jouer plusieurs rôles d'acteur (Role of Actor) dans une organisation. Un acteur est une personne, un programme ou une entité qui peut remplir les rôles à exécuter et être responsable de ces rôles ou être associé dans une certaine mesure avec les activités et procédures. Les attributs ou possibilités d'accès peuvent être associés avec les acteurs et avec les rôles.

Un rôle est défini par : un mot unique (rid) d'identification du rôle, le nom du rôle (name) et la description du rôle. Le rôle d'acteur comprend : l'identification de l'acteur (usrid) et l'identification du rôle (rid).

D'autres paramètres ou attributs et structures peuvent être associés pour créer un modèle organisationnel, destiné à mieux gérer le flux de travail.

L'architecture au niveau du logiciel de traitement transactionnel du flux de travail constituant le système se décompose en un certain nombre de modules (30 à 37 figure 1). Les modules deviennent actifs quand un ouvrage (job) et/ou une activité (activity) est commencé ou achevé. Un module ordonnanceur (30) est activé par un message d'achèvement d'activité, une exception d'activité ou la création d'un ouvrage.

Dans tous les cas, l'ouvrage est démarré par un processus quelconque qui envoie un message du type créer un ouvrage au module ordonnanceur (30) du serveur (3). Les paramètres de ce message comportent la priorité de l'ouvrage, le temps d'expiration, le nom de la procédure et les paramètres d'initialisation.

L'ordonnanceur (30), à travers la faculté de création d'ouvrage, crée l'ouvrage, lui affecte un nombre unique, fixe l'information d'état pour cet ouvrage et effectue différentes initialisations.

Le travail de finalisation est aussi exécuté par l'ordonnanceur (30) à travers la facilité d'achèvement d'ouvrage quand toutes les activités pour un ouvrage ont été exécutées. L'ordonnanceur vérifie tous les prédecesseurs de tous les possibles successeurs d'une activité achevée.

Si l'activité achevée était une activité de décision, alors l'ordonnanceur examine la variable résultant de cette activité. Si un ou plusieurs successeurs sont validés, un message est envoyé à un module répartiteur (33) avec lequel l'ordonnanceur dialogue. L'ordonnanceur consiste en une boucle de programme représentée à l'annexe 2, consistant à lire les messages reçus et à traiter le cas du message.

L'appel de la fonction "WFsvSchedCreateJob" déclenche la vérification que l'état de publication de la procédure est valide et dans l'affirmative crée un nouvel ouvrage pour la procédure identifiée par son identifiant de procédure (pid) avec une priorité déterminée.

L'appel d'une part de la fonction "WFsvSchedStartTask" déclenche la lecture de la table de définition de la procédure avec initialisation d'une tâche et d'autre part de la fonction "WFsvSchedEndTask" déclenche l'achèvement de la tâche avec chargement du vecteur JDV dans le serveur (3).

5 Le module ordonnanceur dialogue avec un module répartiteur (dispatcher 33) et un module signalur (34) (notifier). L'ordonnanceur marque les nouvelles tâches qui sont créées avec l'état "todispatch" et notifie ceci au répartiteur en déposant un message dans une boîte à message (35).

10 Le module répartiteur (33) a trois fonctions principales:
a) affecter à un utilisateur les tâches qui viennent d'être créées par l'ordonnanceur;
b) affecter un nouvel utilisateur à une tâche qui vient d'être rejetée;
c) effectuer la distribution et la circulation d'une activité parmi les utilisateurs.

15 Le module répartiteur (33) est le premier module qui conduit et est conduit par l'acteur et les entités de rôles. Dans sa troisième fonction, pour une activité donnée, le répartiteur (33) établit une correspondance avec un rôle et avec un ou plusieurs acteurs. Si un ou plusieurs acteurs doivent être sélectionnés comme receveur, alors le répartiteur applique les algorithmes de répartition de base effectuant une répartition linéaire, c'est à dire le répartiteur range aléatoirement les receveurs possibles (acteurs) et procède linéairement à travers cette liste. Il délivre le premier ouvrage au premier acteur, le second ouvrage au second acteur et ainsi de suite jusqu'à l'épuisement de la liste. Ceci est ensuite répété continuellement, de façon que tous les acteurs reçoivent approximativement le même nombre d'ouvrage.

20 Dans la première fonction, lorsqu'une première tâche est attribuée, l'état de la tâche dans la table (sched-job) est indiqué "OFFERED" et un message est envoyé à l'utilisateur supposé exécuter l'ouvrage. Dans sa deuxième fonction, lorsqu'un utilisateur rejette une tâche qu'il ne peut exécuter l'état de la tâche est mis à "rejected" dans la table (sched-job) et le répartiteur reçoit un message. Le répartiteur mémorise alors le fait que la tâche a été rejetée et par qui, dans les archives et l'historique de vérification (auditing) et procède comme si la tâche provenait de l'ordonnanceur.

25 Les interfaces de programmation de l'application correspondant à ces trois fonctions sont indiquées dans l'annexe (3).

30 Le module signalur (34) conserve une trace des événements rythmés et signale les receveurs corrects au moment approprié.

35 Deux utilitaires de rythme sont prévus dans le module signalur (34), à savoir la faculté de rappel et la faculté de retardement. Le premier pour, par exemple, permettre d'imposer une butée limite pour l'exécution d'une activité spécifique et envoyer un message d'avertissement quand le temps de butée est expiré, tandis que le second permet de retarder l'expédition d'un message pendant une certaine période de temps pour, par exemple, demander à un acteur de faire un certain travail après un certain retard.

40 Le signalur (34) reçoit également des informations du coordinateur (31) qui prend soin de ce qui se passe dans le système de gestion du flux de travail. Le coordinateur fournit les facilités de contrôle de gestion interne pour garder une trace des événements significatifs et stocker les fichiers de journal affranchis, les façons d'interroger le système sur les différents états (qui fait quoi ? quel est le degré d'avancement d'un ouvrage spécifique ? combien d'ouvrage sont en cours d'exécution ? temps moyen d'exécution d'un ouvrage, temps moyen d'exécution d'une tâche, historique des ouvrages, état courant d'un ouvrage, informations obtenues par un ouvrage déterminé, files pour exécuter une activité donnée au cours d'une procédure donnée, conflits dans un ouvrage donné, etc...) la possibilité d'agir sur l'avancement des ouvrages (suspension, remise en service (resume), annulation (cancel), réattribution (reassign) etc...) et les moyens de garder un historique par archivage.

45 Le répartiteur (33) et le signalur (34) échangent des signaux avec la boîte à messages (35) qui sert à conserver tous les messages asynchrones. Ces messages peuvent être des activités assignées à des acteurs, des messages d'avertissement générant chez le signalur (34), des messages entre acteurs, etc...

50 La boîte à messages est constituée essentiellement dans une base de données relationnelle telle que par exemple la base (e.g. ORACLE de Bull). Le serveur de la boîte à message est un serveur particulier qui gère et mémorise les messages asynchrones.

55 Quand une requête de client est asynchrone, le résultat est mémorisé en tant que message dans la boîte à messages à travers le serveur de la boîte à messages. La seule façon d'accéder à la boîte à messages est à travers le serveur de boîte de messages (35). Ainsi par exemple, quand le répartiteur (33) assigne une tâche à un acteur, un message est envoyé à cet acteur l'informant qu'un sujet de travail est arrivé. Ceci est effectué par le serveur de la boîte de messages qui écrit dans la boîte de messages un message du type "offre de tâche (offer-task)".

Le serveur de boîte de messages est constitué principalement d'une boucle de programmation qui est représentée en annexe 4.

La boîte de messages est constituée principalement de deux tables de données d'une base de données relationnel (371) telle que par exemple "ORACLE". La première appelée MSG-BOX-DESC qui contient la des-

5 ccription des messages tels qu un identifiant unique , le type, etc... La second table est appelée MSG-BOX-BROADCAST qui st utilisée pour mémoriser les destinataires des messages et les autres informations telles que les états de messages. Il st important de noter qu'un message peut être adressé à plusieurs destinataires et qu'en conséquence la deuxième table MSG-BOX-BROADCAST devrait contenir la liste des destinataires tandis qu la description du message st écrite une seule fois dans la table MSG-BOX-DESC.

Enfin un message peut avoir un des trois statuts suivants :

TO-READ : qui signifie que le message est disponible,

READ : qui signifie que le message a été lu,

10 TO-DELETE : qui signifie que le message a été effacé.

Enfin cette boîte à messages (35) reçoit également des signaux de l'administrateur (32). Ce module est utilisé en mode différé pour modeler, créer, tester et accorder les différentes entités du système de base, les procédures, activités, rôles et acteurs.

15 Cette étape peut impliquer l'analyse des structures organisationnelles, des flux d'informations et des procédures de bureau. Les activités "formulaires" et "structures de sécurité" doivent être créées.

Les scénarios de procédures sont écrits par exemple, en utilisant le formalisme d'un logiciel de création d'un réseau de commande d'information ICN (Information Control Net) est décrit dans une autre demande de brevet déposée. Les corps d'activités sont écrits en utilisant une notation fondée sur une base visuelle. Les corps d'activités sont un des véhicules des échanges de données entre les activités du gestionnaire de transaction relationnelles et les différents serveurs extérieurs.

20 Le système comporte également un module de gestion des données (370) qui stocke les tables relationnelles dans la base de données telle que ORACLE pour représenter les définitions, les relations, les attributs de toutes les procédures, activités, ouvrages, rôles et acteurs.

Ces tables sont constituées de tables dynamiques marquées par un astérisque et de tables statiques.

25 Les tables dynamiques sont constituées par les tables SCHED-JOBS, SCHED-STATEP représentée en annexe 8, SCHED-STATUSP représentée en annexe 8, SCHED-TASKP, JDV-JDV et ADV-ADV représentée en annexe 5. Ces tables ont leurs rangées insérées et mise à jour pendant la phase d'exécution de l'ordonnanceur. Normalement il y a un jeu de tables dynamiques pour chaque procédure exécutée sur le système.

Un jeu de table contient toutes les données d'exécution pour tous les ouvrages d'une procédure.

30 Les tables statiques sont constituées par les tables ORGA-ACTOR, ORGA-ROLES, ORGA-R-OFF-ACT, SCHED-PROCEDURES, SCHED- ACTIVITIES, SCHED-PRECEDENCES représentée en annexe 8, JDV-PDDEF (Procédure data definition) représentée en annexe 6, EDA-ORA (external data accessor to ORACLE) représenté en annexe 7. Ces tables statiques contiennent les informations sur les procédures et l'organisation des actions et des rôles.

35 L'ordonnanceur accède à ces tables seulement dans le mode de lecture et se fie à ces informations qu'il mémorise pour l'exécution.

Toutes les tables dynamiques comportent une colonne indiquant l'identificateur de procédure Pid à laquelle les données contenues dans les tables sont affectées. Une colonne nom Jid d'ouvrage est ajoutée à la table SCHED-JOBS et elle sera remplie de façon optionnelle par l'utilisateur à l'initialisation d'un ouvrage.

40 Les informations concernant les données utilisées pour attribuer et répartir les tâches et rôles à l'utilisateur comportent une colonne RoleId dans la table SCHED-TASKP qui sera remplie au moment de la création de la tâche par l'ordonnanceur à partir des informations trouvées dans la table SCHED-ACTIVITIES.

Enfin une table distribution et une table circulation représentées aux annexes 4 et 8 permettent au répartiteur de disposer des éléments pour affecter les activités aux différentes stations des utilisateurs.

45 Avec la table de distribution et le réseau de commande d'informations, il est possible de décrire la circulation d'informations dans un bureau. La distribution décrit la circulation d'une activité entre les différents membres de l'organisation.

La table de distribution comme représentée en annexe 4 comporte un identifiant de distribution Did, un identifiant de procédure (Pid), un identifiant d'activité (Aid), l'information rôle (RoleId), l'information d'identification de l'utilisateur (UserId), une information indiquant que l'identifiant d'utilisateur est au début de la liste de circulation (First) et une information nouvel utilisateur (Next UserId).

La table de circulation est constituée comme représentée en annexe 4 des informations Did, Pid, JobId, Aid pour l'identifiant d'activité, RoleId, UserId, First, Next UserId.

55 La figure 2 représente l'organisation logique des services rendus par un ou plusieurs modules, et leurs interactions avec l'agent utilisateur du système transactionnel qui est l'administrateur, le coordinateur et l'agent.

La table JDV-JDV est constituée comme représentée à l'annexe 5 d'un identifiant d'ouvrage (Jid), d'un identifiant d'activité (Aid), d'un identifiant de tâches (Tid), d'un identifiant de champs dans le vecteur JDV, d'un état (STATUS) qui peut être l'une des valeurs TO-ACCESS, READFWK, READEXTERN, ARCHIVED, d'un identifiant de champs FLD-NAME, d'un mode de passe de champs PASSMODE qui peut avoir les valeurs VA-

5 LUE ou REF pour indiquer qu le champ est transmis par une valeur ou par sa référence, d'un type de champs (FLD TYPE) qui peut être une date, un caractère ou un ntier, d'une longueur en octet de la val ur transmise contenue dans LENGTH, de la valeur du champ si l type de champ est une date contenue dans FLD-DATE, de la val ur du champ si le type d champ st un caractère ou si le mode de passag st REF cont nu dans FLD-KCHAR, de la valeur du champ si l mode d passage est INT contenu dans FLD-INT et d la valeur actuelle du champ désignée dans JDV-PDDEF comme champ de dépendance dans ACCESS-KEY.

10 En addition, la base de données relationnelle sauvegarde plusieurs types de données nécessaires pour l'exécution des activités, telles que les informations spécifiques de procédure, des ouvrages spécifiques contenues dans le vecteur de données d'ouvrage (JDV, Job Data Vector) et les informations spécifiques d'activités 15 contenues dans les vecteurs de données de tâches (TDV, Task Data Vector). Ces éléments de données vectorielles sont des valeurs de données ou pointeur vers des informations stockées sur des répertoires externes d'informations. A chacun des composants dans le vecteur est donné une valeur type et une valeur initiale. La valeur initiale est nulle et est spécifiée par l'événement d'initialisation ou est chargée par le module de création d'ouvrage au démarrage de l'ouvrage. Quand une activité est démarrée, une copie du vecteur de donnée d'ouvrage est rendue disponible pour le client à travers le serveur réseau où la copie du vecteur JDV est chargée vers la station de travail. Certains éléments du vecteur sont lus et certains sont écrits par l'activité lorsqu'elle s'exécute.

20 Quand l'activité est achevée, le vecteur est sélectivement copié ou déchargé vers le serveur (3). De même, les vecteurs de données d'activité stockent les valeurs et les références pour les données spécifiques d'activités. Son format est identique au format du vecteur JDV.

25 Pour réaliser ces différentes actions le système utilise le module de gestion de données (370) qui a pour fonction :

- d'attacher les données appropriées aux ouvrages spécifiques, en créant le vecteur JDV et en démarrant le remplissage de ce vecteur avec les données spécifiques pour ensuite le transmettre à la station de travail (2) quand un utilisateur en a besoin,
- gérer l'intégrité des données et la sécurité en fournissant des mécanismes pour attacher les droits d'accès aux données et des stratégies de verrouillage pour protéger les données contre des accès concurrents,
- retrouver et accéder les données dans les bases de données de l'utilisateur externe en faisant appel à un module EDA d'accès aux bases de données externes. Les données nécessaires à la réalisation de ces fonctions sont stockées dans le fichier définition de données de procédure PDDDEF et dans le fichier EDA-ORA qui sont représentés aux annexes 6 respectivement 7.

30 Les modules (30 à 37) du serveur (3) communiquent à travers le système de communication, comportant les éléments (36,2,17,16) avec un module client consistant dans un ensemble de modules (10 à 15). Ces modules seront décrits dans la suite.

35 Le système serveur (3) communique à travers un réseau (1) et un serveur réseau (364) avec un système client (2) comportant une interface de communication (27) et une architecture à quatre couches, dont la première est le système de messages de flux de travail (26), la seconde couche (24) est constituée par différents 40 jeux d'interface de programmes d'application client (241,242,243,244), d'une troisième couche de service (25) de gestion (251,252,253,254), et d'une quatrième couche (20,21) qui est la couche supérieure d'interface. Les différents modules d'une couche n'ont pas d'interaction entre eux.

45 Le système de messages de flux de travail (26) permet les échanges d'information entre les clients et le serveur.

45 Les informations échangées sont transportées par une structure spécifique appelée "message". Cette couche satisfait les fonctionnalités suivantes : connexion à la couche communication du serveur (364), déconnexion avec la couche communication du serveur et émission et réception de messages.

50 La fonction connexion permet d'établir un lien permanent avec le serveur (36). Une station de travail peut supporter une seule liaison. Pour cela, tous les messages qui sont échangés par des fonctions d'émission et de réception passent à travers ce lien.

55 Un message est composé d'une partie enveloppe et d'une partie donnée. La partie enveloppe contient l'émetteur, le destinataire et les informations d'état, tandis que la partie donnée contient le contenu du message.

Le système de messages (26) a aucune connaissance au sujet du contenu du message, ni de l'enveloppe et encore moins des données. Ce module envoie la partie enveloppe et le nom du fichier qui contient la partie donnée à travers le lien réalisé avec le serveur, tandis que le fichier de la partie donnée est déjà sur le système de fichier partagé (372) qui est partagé avec toutes les stations de travail et le serveur.

Les interfaces de programmation d'application des clients regroupent toutes les interactions entre la station de travail et le serveur. Ces interfaces sont classées selon les activités suivantes : administration, constitue

une interface (254), coordination constitue un interfac (254), création d'ouvrage constitu un interface (253), l'exécution d'activité constitue une autre interface (252), les tâches constituent un dernière interfac (251).

5 Toutes ces interfaces d programmation sont générées par un générateur automatique de cod s pour un description donné d s fichi rs en entrée. Ceci assur la cohérence entr les interfaces d programmation client et du serveur.

10 La couche d'interface de programmation administration (244) définit les données qui sont classées selon les concepts et les données définis pour le système de traitement transactionnel. Elle permet la définition des procédures, des activités, des acteurs, des rôles et de l'affectation ou de la désaffectation d'un rôle à un acteur.

15 Les interfaces de coordination (244) permettent la commande de l'exécution d'ouvrage et sont constituées des éléments suivants : état d'un ouvrage, avancement d'un ouvrage, historique d'un ouvrage, qui fait quoi, nombre d'ouvrage actif, suspension d'un ouvrage, reprise d'un ouvrage, annulation d'un ouvrage.

20 Le jeu d'interfaces client (243) pour la création d'un ouvrage est constitué des interfaces qui participent à la création d'un nouvel ouvrage dans une procédure déterminée. Le jeu d'interface exécution d'activité n'est pas défini.

25 Le jeu d'interface tâche (241) donne à un acteur les facilités pour exécuter les tâches ordonnées et réparties. Ces facilités sont classées de la façon suivante : lecture d'une tâche offerte, acceptation d'une tâche offerte, recherche du vecteur de donnée d'ouvrage correspondant dans le serveur, achèvement d'une tâche et restitution du vecteur de donnée d'ouvrage réactualisé au serveur, réorientation d'une tâche à un autre acteur ou activité.

30 La couche de gestion des services regroupe des modules (251 à 254) qui gèrent les différents services proposés à l'acteur. Chaque module fournit un jeu bien défini de service à l'acteur, traite et maintient les données utiles localement et appelle les interfaces de programmation d'application client appropriées et respectives (241 à 244) pour fournir le service souhaité.

35 Un module gestion administration et coordination (242) fournit à l'acteur les services pour définir les concepts et données de flux de travail. Ceci inclut les liens avec les interfaces appropriées et le support des définitions de processus. La gestion de coordination envoie au serveur les requêtes de coordination venant de l'interface utilisateur correspondante et traite les réponses reçues.

40 Le module de gestion de création d'ouvrage (253) se résume à transmettre les créations d'ouvrage, les requêtes de création d'ouvrage au serveur et peut être capable de fournir une cession de création d'ouvrage pendant que l'acteur prépare les données d'initialisation correspondant à la procédure de définition de données PDDEF et lance la création d'ouvrage effective sur le serveur.

45 Le module de gestion de l'exécution d'activité (252) est dédié à réaliser l'exécution de la partie corps de l'activité. Cette exécution est requise par le module de gestion de tâche (251) à travers l'interface utilisateur (20). Une telle exécution peut être considérée comme une cession de travail pendant laquelle l'acteur exécute les actions correspondantes.

50 En dehors de ces actions qui sont une partie de l'exécution normale d'une activité, l'acteur a la possibilité de suspendre, annuler et bien entendu terminer un cession de travail.

55 Dans ce but le module de gestion d'exécution d'activité est pourvu des possibilités :

- de lecture du fichier de description d'activité ;
- de lecture et d'écriture des données d'ouvrage utilisées dans l'activité ;
- de lecture et d'écriture des données de tâches correspondantes utilisées dans l'activité et de commande des droits d'accès sur les données d'ouvrage ;
- de lancement du formulaire de présentation approprié de l'activité et de l'exécution des actions correspondantes à l'activité d'une façon guidée.

60 Un module de gestion de tâches (151) est dédié à gérer l'aspect multi-tâche d'un acteur puisque le système de traitement transactionnel donne la possibilité à un acteur de traiter plusieurs tâches en même temps. Pour cela, ce module maintient localement une mémoire des tâches et fournit la possibilité de sauvegarder, de retrouver cette mémoire et d'effacer des tâches de cette mémoire. Ce module comporte un jeu d'opération pour exécuter normalement les tâches qui sont les suivantes :

- réception d'une tâche offerte,
- acceptation d'une tâche offerte,
- recherche des données correspondantes du vecteur d'ouvrage sur le serveur,
- achèvement d'une tâche,
- restitution du vecteur d'ouvrage mis à jour au serveur,
- réorientation d'une tâche vers un autre acteur ou un autre activité,
- appel de l'activité d'exécution d'une tâche correspondante par l'interfac utilisateur,
- traitement et maintient local des données correspondant aux tâches et leur historique.

Enfin un module de gestion de note, non représenté, est également prévu pour fournir à l'acteur la facilité de création de note, d'attachement de note, de lecture et d'écriture, réception et émission de note, traitement local et maintenance de notes envoyées et reçues.

5 La couche supérieure est constituée de la couche d'interface qui comporte une interface coordinateur et administrateur (21) et une interface utilisateur (20).

10 L'interface administrateur et coordinateur (21) comporte les éditeurs de concept et de données et la commande des ouvrages. L'interface utilisateur comporte les logiciels, les parties permettant l'utilisation interactive du système de traitement transactionnel, l'exécution d'activité guidée par un formulaire et l'utilisation automatique du système de traitement transactionnel. Ces interfaces sont lancées soit par l'acteur lui-même, soit par le module de gestion (251).

15 L'interface utilisateur permet à l'acteur une participation et une contribution au travail de bureau. À travers celle-ci, l'acteur est capable d'accomplir les tâches qui lui sont attribuées, de démarrer un nouvel ouvrage et d'échanger des messages avec les autres acteurs. L'acteur est identifié par son nom et son mot de passe dans le processus de connexion.

20 Au cours du fonctionnement, les événements de création d'ouvrage incitent le module ordonnanceur (scheduling) à démarrer un nouvel ouvrage. Ainsi l'ordonnanceur peut démarrer une première activité. Ensuite un acteur (un utilisateur spécifique) est alloué par le répartiteur à la première activité et l'ouvrage est envoyé à la station de travail du client approprié. Quand la station de travail achève l'activité, l'ouvrage est retourné au serveur où l'ordonnanceur met à jour l'état de l'activité et trouve une nouvelle activité. L'ordonnanceur appelle le répartiteur pour attribuer les acteurs en fonction des rôles et ensuite l'information d'ouvrage et les demandes d'exécution sont envoyées à la machine client appropriée pour exécuter l'activité suivante.

25 Tous ces états, activité, rôle, utilisation d'acteur, attribution d'information sont enregistrés par la facilité de vérification (auditing) du module coordinateur une fois les attributions faites. Ce cycle se poursuit jusqu'à ce que la dernière activité soit achevée et à ce moment-là, l'ordonnanceur procède à l'achèvement de l'ouvrage.

30 Le module de gestion de l'exécution d'activité (252) contrôle et supporte les exécutions d'activités, en exécutant le corps de programme correspondant à l'activité demandée. Ce corps peut accéder à des données externes et avoir accès à des données spécifiques de l'ouvrage ou des données spécifiques d'activités. Puisque cette activité est fréquemment interactive, le module d'interface utilisateur est appelé par ce corps. Les documents et les formulaires sont affichés et manipulés par l'utilisateur comme une partie de cette activité.

35 Quand l'activité est achevée, un message est envoyé au logiciel du serveur indiquant le nouvel état. Comme un client peut travailler sur plus d'un ouvrage simultanément, le module gestionnaire de tâches (251) maintient les statuts des ouvrages et présente les différents ouvrages et leurs statuts à l'utilisateur. À chaque station de travail, l'information appropriée est accessible et toutes les informations sont maintenues quand un utilisateur éteint sa station de travail. Des rôles multiples peuvent être résidents sur une seule machine client.

40 Les interactions entre les modules sont facilitées par le sous-système de communication (27,1,36) qui autorise les échanges de messages entre les modules du système et aussi entre les utilisateurs. Ce système supporte à la fois des communications synchrones (ex : fetchtask) et des communications asynchrones (ex : sendto). Les interactions de ces communications sont mises en œuvre par les différentes couches du protocole de communication.

45 Dans une variante de réalisation le système de traitement transactionnel peut être utilisé avec une interface utilisateur telle que celle décrite ci-après qui permet une grande facilité d'utilisation pour l'utilisateur.

Cette interface est basée sur des documents, des icônes et des formulaires et est constituée sous forme de trois dossiers qui peuvent être accédés et manipulés par le système de traitement transactionnel.

50 Ce sont les dossiers, nouvelles tâches (new tasks), tâches en cours (on going tasks) et tâches achevées (completed tasks). Les tâches qui sont les travaux à effectuer peuvent être conceptualisées comme des "instances d'activité".

Les éléments dans le dossier "nouvelles tâches" sont des ouvrages qui sont dans l'état validé prêt à être exécutés par ce rôle. En ouvrant le dossier, l'utilisateur peut demander et initier de nouvelles tâches. Cette action initiation, fait passer l'état au mode actif, déplace l'icône de la tâche dans le dossier "tâches en cours" et ouvre le premier formulaire associé avec la tâche sur l'écran utilisateur.

55 En ouvrant le dossier "tâches en cours", l'utilisateur peut visualiser l'avancement des tâches. L'état, la priorité et l'urgence de chaque tâche est également visualisable. De nombreuses tâches peuvent être actives avec aucun, un ou plusieurs de leurs formulaires de tâches ouverts simultanément sur la même station.

Le dossier "tâches achevées" est le moyen pour l'utilisateur d'interroger le système sur l'état courant, l'historique passé ou les défauts de comportement des ouvrages.

Le dossier "notes reçues" contient les messages expédiés par les autres utilisateurs du système de traitement transactionnel ou par les modules du système et qui ont été reçus. Ceci permet des communications informelles et flexibles entre les utilisateurs du système.

En général, les informations spécifiques de tâches, les outils et les accès externes sont représentés comme des éléments, icônes et menus sur le formulaire de tâches.

5 Cette interface (20) comporte un logiciel de présentation et de constitution des dossiers. Ce logiciel est en relation par un liaison du terminal avec une souris qui permet de provoquer des événements extérieurs. La souris a été choisie du type à deux boutons, le bouton de gauche étant affecté par convention aux sélections de zones ou d'objets graphiques à l'écran et aux appels d'éditeur, le bouton de droite étant affecté à des opérations spécifiques telles que la manipulation de certains menus. Les événements extérieurs provenant de la souris sont traités par une interface souris préprogrammée incorporée au logiciel et susceptible en outre de reconnaître l'appui (down), le relâchement (up) d'un bouton, la traîne (drague) de la souris et des événements logiques tels que l'entrée du pointeur souris dans la fenêtre. A l'écran la position du pointeur de la souris est repérée par une petite flèche dirigée vers le haut.

10 Bien entendu, sans sortir du cadre de l'invention, la souris pourrait être remplacée par tout autre dispositif de pointage tel qu'un crayon optique ou une table graphique. Enfin en complément de l'interface souris, on prévoit également un interface pour un clavier programmé autant pour les touches de caractères que celles de contrôles et celles de pointeurs, de façon à pouvoir entrer dans le mode édition les informations qui correspondent aux différentes zones des différentes fenêtres.

15 Le logiciel de présentation permet l'affichage des fenêtres et l'exécution des actions sélectionnées sur le menu de la fenêtre à la suite du déclenchement d'un événement extérieur, tel que par exemple, l'actionnement d'un bouton souris. La première fenêtre affichée après la connexion avec le serveur est la fenêtre (40, fig 4).

20 Cette fenêtre (40, fig 4) comporte une barre de titre (41) en dessous de laquelle se trouve une barre menu (42) permettant de sélectionner un des objets se trouvant au menu à savoir, dossier (420), ouvrage (421), note (422) et aide (423). Chacun des objets sélectionnés faisant apparaître la possibilité d'un choix d'attribut, par exemple pour l'objet dossier (420), le choix entre l'attribut nouveau (4200), en cours (4201), réalisé (4202), ou note reçue (4203).

25 Ce choix permet en le validant de changer de fenêtre et de faire apparaître une des autres fenêtres "dossier en cours" (50 fig 5) "dossier exécuté" (60 fig 6) ou "note reçue" (70 fig 7).

30 Pour l'objet ouvrage (421), le choix consiste entre les fonctions "accepte" (4210) pour accepter une tâche, action qui est implicite si la tâche est démarrée, "traitement" (4211) pour démarrer ou continuer l'exécution d'une tâche, "feuille de propriété" (4212) pour voir la fiche qui montre plus d'informations au sujet de la tâche de la liste, la fonction "commentaire" (4213) pour voir les commentaires concernant la tâche courante et les commentaires édités à l'étape précédente et pour éditer éventuellement de nouveaux commentaires. La fonction "envoi" (4214) pour envoyer toutes les données d'activité à un acteur, un rôle ou une activité. Cette fonction pouvant être aussi utilisée pour rejeter une tâche. La fonction "orientation au suivant" (4215) pour préciser que la tâche est achevée et peut être envoyée à la prochaine étape de la procédure. La fonction "création" (4216) pour créer un nouvel ouvrage.

35 Dans le menu "ouvrage" on constatera que la fonction "orientation au suivant" apparaît en caractère moins gras que les autres fonctions, car cette fonction ne peut être sélectionnée lors de l'affichage de la fenêtre "nouveau dossier" indiquée par la barre de titre (41).

40 Le menu "note" (422) comporte la fonction "lecture" (4220) pour visualiser une note reçue ou envoyée. La fonction "expédition" (4221) pour éditer et envoyer une note. La fonction "annule" (4222) pour effacer de la liste une note reçue ou envoyée, la fonction "liste des notes envoyées" (4223) pour être capable de visualiser la liste des notes qui ont été envoyées.

45 Le menu "aide" (423), comporte les fonctions "index", "clavier", "commandes", "procédures", "utiliser l'aide", "à propos de", qui sont des fonctions classique et ne seront pas décrites en détail.

50 Dans le menu "note" (422) on constate également que la fonction "lecture" (4220) et "annuler" (4221) ne sont pas en caractère gras comme les autres (envoi, liste des notes), ce qui indique que ces fonctions (4220, 4221) ne peuvent être sélectionnées lors de l'affichage de la fenêtre (40) "nouveau dossier". Cette fenêtre "nouveau dossier" se complète d'un cadre (43) dans lequel une colonne "ouvrage" (430) permet d'indiquer l'identifiant de l'ouvrage, une colonne (431) "activité" qui permet d'indiquer le nom de l'activité, une colonne (432) qui permet d'indiquer à quelle date l'activité doit être terminée et une colonne commentaire dans lequel un signe apparaît pour indiquer que des commentaires sont attachés à l'ouvrage ou non. Enfin une étiquette (434) apparaît en face de l'ouvrage ou de l'activité créée pour indiquer son urgence ou non.

55 Lorsque l'on est dans la fenêtre et dans le menu "nouveau dossier", la ligne (44) de bouton de sélection de l'attribut des dossiers fait apparaître le bouton de sélection nouveau (4220) avec un fond différent des boutons de sélection "en cours" (4201) "exécuté" (4202) ou "notes reçues" (4203).

La fenêtre comporte également des boutons de commandes (45) qui permettent de lancer directement les fonctions "accepte" (4210), "traitement" (4211), "commentaire" (4213), "feuille de propriété" (4212) du menu "ouvrage" (421).

La fenêtre "dossier en cours" (fig 5) comporte les mêmes menus et les mêmes éléments sauf pour la barre de bouton de commande (55) qui fait apparaître les fonctions "orientation vers le suivant", "traitement", "feuille de propriété" et "commentaire". A la différence également de la fenêtre de la figure 4, cette fenêtre ne permet pas la sélection de la fonction "accepte" (4210).

La fenêtre "dossier exécuté" représentée à la figure 6 comporte outre la barre titre et la barre menu dans le cadre (63), en plus des colonnes "ouvrage" et "activité", une colonne (66) permettant d'afficher la date d'achèvement de l'ouvrage ou de l'activité et en bas du cadre, une ligne de bouton de commande (65) constituée par les commandes "feuille de propriété", "commentaires" et "effacement", ces fonctions sont d'ailleurs les seules qui peuvent être sélectionnées dans le menu "ouvrage".

Enfin la figure 7 représente la fenêtre des "notes reçues" qui comporte la barre titre, la barre de menu et un cadre (73) qui permet dans une colonne (731) d'afficher "l'expéditeur", dans une colonne (732) de mettre la date de réception de la note et dans une colonne (733) d'afficher le sujet de la note.

Cette fenêtre comporte également une barre (75) constituée d'un seul bouton de commande "lecture Note" permettant la lecture de la note.

La sélection de la fonction "feuille de propriété" (4212) dans une fenêtre "dossier nouveau" (fig 4) ; dossier en cours (fig 5) ; "dossier exécuté" (fig 6) fait apparaître la fenêtre "propriété de l'ouvrage" représentée à la figure 8. Cette fenêtre permet d'appeler une fonction d'édition et comporte un premier champ (81) permettant d'inscrire le nom de l'ouvrage et une étiquette (82) faisant apparaître le terme "urgent" si cet ouvrage a été désigné comme urgent.

Un deuxième champ (83) permet d'indiquer la date à laquelle l'ouvrage doit être terminé, un troisième champ (84) permet d'indiquer le nom de la procédure auquel l'ouvrage est rattaché, un cinquième champ (85) permet d'indiquer le nom du responsable pour l'ouvrage et une fenêtre déroulante (86) permet d'afficher la liste des documents attachée à l'ouvrage. Une boîte (87) permet d'afficher 4 champs concernant l'activité courante, un premier (870) indiquant le nom de l'activité, un second (871) indiquant l'état de la tâche de l'activité courante, un troisième (872) indiquant le responsable de la tâche et un quatrième (873) donnant la date à laquelle la tâche doit être exécutée.

Deux flèches non représentées dans la barre des boutons de commande (880 à 883) l'une avec l'étiquette précédente, l'autre avec l'étiquette suivante, permettent de voir respectivement la feuille de propriété d'ouvrage précédent et la feuille de propriété d'ouvrage suivant.

Le bouton de commande "accepte" (880) n'est utilisable que quand l'état de la tâche est nouveau. Un deuxième bouton de commande "traitement" (881) n'est utilisable que quand l'état de la tâche est accepté, suspendu ou nouveau. Un troisième bouton de commande "commentaires" (881) ouvre la fenêtre commentaire qui est décrite en relation avec la figure 9. Un quatrième bouton de commande "fermeture" (882) ferme la fenêtre de "propriété d'ouvrage" (80) boîte et un cinquième bouton de commande "Aide" (883) permet d'obtenir une assistance.

La fenêtre de "commentaire" (90) comporte une barre titre (900), une barre menu (91) permettant d'appeler la fonction d'édition ou d'impression, un champ "ouvrage" (92) dans lequel le nom de l'ouvrage peut être reporté dans la boîte d'édition (920) et deux boîtes de texte (93,94) étiquetées "commentaires précédents" et "nouveaux commentaires". Un bouton de commande "OK" (921) permet de fermer la boîte en effectuant l'enregistrement du nouveau commentaire s'il y en a eu.

Le menu (91) comporte un menu "d'impression" permettant de lancer les fonctions suivantes : "commentaires précédents", "nouveaux commentaires", "sélection de tous", et un menu d'édition permettant de lancer les fonctions : "couper", "copier", "coller", "effacer".

La figure 10 représente la fenêtre qui est affichée sur l'écran de la station de l'utilisateur lorsque celui-ci sélectionne la fonction "émission d'ouvrage". Cette fenêtre comporte une barre de titre "émission d'ouvrage" (100) et un champ (101) permettant d'indiquer le nom de l'ouvrage, une partie (102) concernant les récepteurs, une partie (106) concernant la rémission de l'ouvrage, une boîte de texte étiquetée "nouveaux commentaires" (105) qui comprendra tous les commentaires de l'émetteur et dans laquelle l'émetteur sera capable d'ajouter des nouveaux commentaires destinés au récepteur, un bouton de contrôle (107) assurant la fonction "d'émission retardée jusqu'au" suivi d'une boîte de texte (108) dans laquelle l'utilisateur peut entrer la date à laquelle il souhaite que la tâche soit envoyée, les boutons de commande "émission" (1090) dont la fonction dépend ou non du fait qu'il y a plusieurs récepteurs, s'il y a un seul récepteur avec un clic sur le bouton de commande, l'émission est achevée, s'il y a plusieurs récepteurs, un clic sur le bouton "envoi" ouvre la boîte de dialogue "mode d'émission" décrite ci-après en liaison avec la figure 11, un bouton de commande "annule" (1091) pour fermer la boîte "émission d'ouvrage" sans effet.

La partie concernant le récepteur est composée de boîtes combinées étiquetées "role" (1020), "nom" (1021), "activité" (1022) dans laquelle l'utilisateur peut choisir un élément ou écrire ce qu'il souhaite. La valeur défaut n'est pas définie, seulement un des trois paramètres du récepteur doit être rempli. Si l'utilisateur n'

remplit pas au moins un des paramètres, la commande "envoi" ne sera pas valable.

Une boîte de liste dont chaque ligne est complétée avec les valeurs des boîtes de combinaison, un bouton de commande "suivant" (1024) pour aller à la ligne suivante, un bouton de commande "efface" (1025) pour effacer une ligne de la liste, un bouton de commande "insère" (1026) pour insérer une ligne libre entre la ligne sélectionnée et la ligne du dessous.

La fenêtre 11 du mode d'émission permet par sa boîte de dialogue comportant deux boutons d'options, un premier (111) de distribution séquentielle, d'envoyer l'ouvrage à chaque personne de la liste dans un mode séquentiel et dans l'ordre de circulation correspondant avec un ordre de circulation correspondant à l'ordre de la liste, un deuxième bouton (112) de distribution simultané dans lequel l'ouvrage sera envoyé à chaque personne de la liste dans un mode parallèle.

Cette boîte de dialogue comporte également les boutons de commande "OK" (113) pour confirmer le choix et achever l'émission, un bouton de commande "annule" (114) pour fermer la boîte et un bouton de commande (115) pour demander une assistance.

La fonction "création d'ouvrage" est représentée à la figure 12 par une fenêtre (12) comportant une barre de titre "création d'ouvrage" (120) et un champ (121) permettant de rentrer dans un pavé d'édition de texte le nom de l'ouvrage, un deuxième champ "référence de procédure" permettant de rentrer dans un pavé (122) le nom de la procédure et dans un pavé (123) une étiquette, un troisième champ "responsable de l'ouvrage" qui permet de rentrer dans un quatrième pavé (124) le nom du responsable, un cinquième champ "date à laquelle l'ouvrage doit être exécuté" comportant un cinquième pavé (125) permettant de rentrer la date, un sixième champ (126) permettant d'afficher la liste des documents existants et les boutons de commande "OK" de validation (127), d'annulation (128), et d'assistance (129).

La fonction "émission de notes" est représentée par la figure 13 et fait apparaître une fenêtre (13) comportant une barre de titre (130), un premier champ (131) permettant de faire apparaître le récepteur, un deuxième champ (132) permettant de faire apparaître la date d'émission, un troisième champ (133) permettant de faire apparaître le sujet de la note, un ensemble de deux boutons d'option permettant de choisir entre la distribution séquentielle par un bouton (134) et la distribution simultanée par un bouton (135). Une boîte (136) permet de sélectionner les noms des notes à envoyer parmi une liste figurant dans cette boîte. Une flèche (137) "précédente" et une autre flèche (138) "suivante" permet de sélectionner dans la liste la note précédente ou la note suivante. Enfin les boutons de commande "récepteur" (1390) permet d'appeler la fonction permettant l'ouverture de la liste des récepteurs qui est représentée à la figure 14, un bouton (1391) permet d'effacer une note, un bouton (1392) permet de fermer la fenêtre "émission de notes", et un bouton (1393) permet de faire appel à une aide.

La boîte (14) "liste des récepteurs" comporte un champ (141) indiquant la date d'émission et une boîte de texte comportant la liste des récepteurs qui est composée de trois colonnes, une première (142) indiquant le rôle, une deuxième (143) indiquant le nom et une troisième (144) l'activité. Un bouton de commande "fermeture" (145) permet de fermer la fenêtre 14.

La fonction "notes reçues" est attachée à la sélection de la fonction "note lecture" (4220) et permet l'affichage d'une fenêtre identique à celle de l'émission à part que le titre est "réception de notes", l'émetteur remplace le récepteur, les deux boutons d'option pour l'émission sont supprimés et la possibilité de documenter une liste de récepteurs par un bouton de commande a été également supprimé puisque cette fonction n'est pas nécessaire dans la réception de notes. Les autres éléments à savoir la boîte de texte avec barre de défilement horizontale et verticale, les flèches "précédent" "suivant", la commande "effacement", "fermeture" et "assistance" sont conservés.

Enfin la fonction "liste des notes envoyées" sélectionnée fait apparaître une fenêtre (15) représentée à la figure 15 dans laquelle une boîte de listage (151) avec les colonnes suivantes, "date d'émission", "sujet", "récepteur" "émis" permet de faire apparaître dans chaque ligne la date d'émission, le sujet, le récepteur vers lequel la note a été émise, et le fait de savoir si la note a été émise séquentiellement ou simultanément. Cette fenêtre comporte également les boutons de commande d'affichage de la liste des récepteurs (152), de lecture de la note (153), d'effacement (154), de fermeture (155) et d'assistance (156).

Les fonctions et les présentations décrites ci-dessus réalisées par le logiciel de présentation et de constitution de dossier qui fait partie de l'interface utilisateur facilitent grandement l'utilisation du système de traitement transactionnel et améliorant son ergonomie.

ANNEXE 1

5

10

Activité du corps structure

[ProcedureId ActivityId]

A0 = ActivityPreScript

Z0 = ActivityPostScript

20

A1 = PresScript

Z1 = PostScript

25

AN = PreScript

30

ZN = PostScript

PR = ActivityPresentation

35

ADV = 'File name'

40

45

50

55

ANNEXE 2

5 While(1)

()

10 Read Message

15 switch(Message type)

()

```

15        case Login    :    ReturnCode = WFsvSchedConnect();
          case CreateJob :    ReturnCode = WFsvSchedCreateJob();
          case StartTask :    ReturnCode = WFsvSchedStartTask();
          case EndTask    :    ReturnCode = WFsvSchedEndTask();

```

20

25 1. WFsvSchedCreateJob(

| | |
|--------------|------------|
| tWFTaskCoord | *TaskCoord |
| long | *InitUsrId |
| char | *Priority |

```

30        IN :    TaskCoord.pid
          OUT :   TaskCoord.jid
          OUT :   TaskCoord.aid
          OUT :   TaskCoord.tid
          IN :    Priority

```

35

40 2. WFsvSchedStartJob(

| | |
|--------------|------------|
| tWFTaskCoord | *TaskCoord |
|); | |

```

45        IN :    TaskCoord.pid
          IN :    TaskCoord.jid

```

45

50 2. WFsvSchedEndTask(

| | |
|--------------|------------|
| tWFTaskCoord | *TaskCoord |
| tWFjDV | *JDV |
|); | |

55

```

55        IN :    TaskCoord.pid
          IN :    TaskCoord.jid
          IN :    TaskCoord.aid
          IN :    TaskCoord.tid
          IN :    JDV

```

ANNEXE 3

5 1. WFsvDispAssign(

tWFTaskCoord *TaskCoord

10);

IN : TaskCoord.pid

15 2. WFsvDispRandAssign(

tWFTaskCoord *TaskCoord

20);

IN : TaskCoord.pid

25 IN : TaskCoord.jid

IN : TaskCoord.aid

IN : TaskCoord.tid

30 3. WFsvDispLeastLoadAssign(

tWFTaskCoord *TaskCoord

35);

IN : TaskCoord.pid

IN : TaskCoord.jid

40 IN : TaskCoord.aid

IN : TaskCoord.tid

45 4. WFsvDispStartTask(

tWFTaskCoord *TaskCoord

50 tWFjDV *JDV

);

55 IN : TaskCoord.pid

IN : TaskCoord.jid

IN : TaskCoord.aid

IN/OUT: TaskCoord.tid

OUT : JDV

ANNEXE 3(suite)

5

5. WFsvDispGiveTaskTo(

10

tWFTaskCoord

TaskCoord

long

SenderId

15

long

ReceiverUserId

tWFjDV

JDV

20

);

25

IN : TaskCoord.pid
 IN : TaskCoord.jid
 IN : TaskCoord.aid
 IN : TaskCoord.tid
 IN : SenderUserId
 IN : ReceiverUserId
 IN : JDV

30

6. WFsvDispSendCopyOfTaskTo(

35

tWFTaskCoord

TaskCoord

long

SenderId

40

long

ReceiverUserId

tWFjDV

JDV

45

);

50

IN : TaskCoord.pid
 IN : TaskCoord.jid
 IN : TaskCoord.aid
 IN : TaskCoord.tid
 IN : SenderUserId
 IN : ReceiverUserId
 IN : JDV

55

5

ANNEXE 4

10

| | | | | |
|----|--------------------------|---------------|-----------------------------|--|
| | disp_distribution | Did | Number(5) | //Distribution Id |
| | | Pid | Number(4) | //Procedure Id |
| 15 | | Aid | Number(5) | //Activity Id |
| | | RoleId | Number(5) | |
| 20 | | UserId | Number(5) | // A UserId that must fill the role. Can be left // NULL or filled if known at definition time. |
| | | First | char(1) | // True if UserId is at the beginning of the // circulation list. |
| 25 | | | NextUserId Number(5) | //Next User on the list |

30

| | | | | |
|----|-------------------------|---------------|-----------------------------|--|
| | disp_circulation | Did | Number(5) | //Distribution Id |
| 35 | | Pid | Number(4) | //Procedure Id |
| | | JobId | Number(6) | //JobId |
| 40 | | Aid | Number(5) | //Activity Id |
| | | RoleId | Number(5) | |
| 45 | | UserId | Number(5) | // A UserId that must fill the role. Can be left // NULL or filled if known at definition time. |
| | | First | char(1) | // True if UserId is at the beginning of the // circulation list. |
| 50 | | | NextUserId Number(5) | //Next User on the list |

55

ANNEXE 5

5

jdv_jdv table description

| 10 | Name | Null? | Type |
|-----------|----------|-------------|------------|
| 15 | JID | NOT NULL | NUMBER (6) |
| AID | NOT NULL | NUMBER (8) | |
| 20 | TID | NOT NULL | NUMBER (7) |
| FLDID | NOT NULL | NUMBER (10) | |
| 25 | STATUS | | CHAR (10) |
| FLDNAME | | | CHAR (59) |
| READ | | | CHAR (1) |
| 30 | WRITE | | CHAR(1) |
| PASSMODE | NOT NULL | | CHAR (5) |
| 35 | FLDTYPE | NOT NULL | CHAR (15) |
| LENGTH | NOT NULL | | NUMBER (5) |
| 40 | FLDDATE | | DATE |
| FLDCHAR | | | CHAR (25) |
| 45 | FLDINT | | NUMBER |
| EDATYPE | | | CHAR (255) |
| 50 | EDAINDEX | | NUMBER (8) |
| ACCESSKEY | | | CHAR (255) |

55

ANNEXE 6

5

jdv_pddef table

10

| | Name | Null? | Type |
|-----------|------------|------------|-------------|
| 15 | PID | NOT NULL | NUMBER (5) |
| AID | NOT NULL | NUMBER (8) | |
| 20 | FLDID | NOT NULL | NUMBER (10) |
| FLDNAME | | CHAR (59) | |
| 25 | ARG | | CHAR (1) |
| READ | | CHAR (1) | |
| WRITE | | CHAR(1) | |
| 30 | DEPENDANCE | | NUMBER(10) |
| PASSMODE | NOT NULL | CHAR (5) | |
| 35 | FLDTYPE | NOT NULL | CHAR (15) |
| MAXLENGTH | | NUMBER (5) | |
| 40 | EDATYPE | NOT NULL | CHAR (10) |
| EDAINDEX | NOT NULL | NUMBER (8) | |

45

50

55

5

ANNEXE 6(suite)

Where :

10

Pid Procedure Id

15

Aid Activity Id. If 0 the field is global to the procedure (and does not have to be redefined for each activity). If it has a positive value it defines the field locally for this activity.
The local definition preempts the global one.

20

FldId JDV Field Id

25

FidName String identifier of the field (For conviviality).

30

Arg "A" a value is mandatory to initiate the job "--" if not.

35

Read "R" if the field is in read access mode within <Pid,Aid>, "--" if not.**Write** "W" if the field is in write access mode within <Pid,Aid>, "--" if not.

40

Dependance Field Id of another Field that will determine the value of the current field.
(For example : Customer Id depends on Customer Name)

45

Passmode "Value" if the field is passed by value, "Ref" if passed by reference.

50

FldType (See discussion below).

55

FldType must be known on the Workstation by the Activity management Module which executes the activity scripts (See flowWork WorkStation Specification). Possible values for FldType can be :

60

Date, Char, Int : Date , string or integer value.

Form : the JDV field can be the frame of a form.

65

IWsRefCabinet : A string containing a server name and the cabinet name.

IWsRefFolder : (Folder Location. For example : /gnomo/GMF/Clients)

IWsStoreDoc : (Storage index reference)

70

MaxLength Number of digits authorized for display (used for form fields only).

75

EdaType Accessor type. Can be "EDA_ORA", "EDA_IWS" for Version1. Actually points to a table.

80

Edaindex Tuple identifier in the EdaType table. The tuple in the EdaType table is the data accessor (see below) which references the actual piece of information corresponding to the JDV Field.

5

ANNEXE 7

10

eda_ora table description

15

| Name | Null? | Type |
|------------|----------|------------|
| EDAINDEX | NOT NULL | NUMBER (8) |
| REPOSITORY | | CHAR (30) |
| LOGIN | | CHAR (30) |
| SECTION | | CHAR (30) |
| SELECTOR | | CHAR (30) |

20

25

30

35

40

45

50

55

5

ANNEXE 7(suite)

10

Edaindex Tuple identifier referenced by a field definition in the Procedure Data definition (jdv_pddef) table.

15

Repository name of the server which is host of the User Data Base.

Login the login under which the EDA_ORA process will connect to the foreign Data base.

20

Section and Selector correspond respectively to what is known in the data base as table and column.

25

Thus data base accessor consists of :

Repository,

Login

Section

Selector

35

Dependance

AccessKey

40

Eventually a query can be made up of these elements :

CONNECT Repository/Login

45

SELECT Selector FROM Section

WHERE Dependance = AccessKey

50

55

ANNEXE 8

5

SQL> desc sched_precedence;

10

| Name | Null? | Type |
|------------|----------|------------|
| INDEX_ID | NOT NULL | NUMBER (8) |
| PREV_AID | NOT NULL | NUMBER (8) |
| NEXT_AID | NOT NULL | NUMBER (8) |
| OUT_BRANCH | NOT NULL | NUMBER (3) |
| IN_BRANCH | NOT NULL | NUMBER (3) |
| PID | NOT NULL | NUMBER (5) |
| TASK_BACK | NOT NULL | NUMBER (7) |
| NEXT_TYPE | | CHAR (10) |

20

SQL> desc sched_statusp;

25

| Name | Null? | Type |
|----------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| QUEUE | NOT NULL | NUMBER (3) |
| INDEX_ID | NOT NULL | NUMBER (8) |

30

SQL> desc sched_statep;

35

| Name | Null? | Type |
|----------------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| AID | NOT NULL | NUMBER (8) |
| STATE | NOT NULL | CHAR (10) |
| TOKENS_CURRENT | NOT NULL | NUMBER (3) |
| TOKENS_PAST | NOT NULL | NUMBER(3) |

45

SQL> desc sched_dyn_prec;

50

| Name | Null? | Type |
|------------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| INDEX_ID | NOT NULL | NUMBER (8) |
| PREV_AID | NOT NULL | NUMBER (7) |
| NEXT_AID | NOT NULL | NUMBER (7) |
| OUT_BRANCH | NOT NULL | NUMBER (3) |
| IN_BRANCH | NOT NULL | NUMBER (3) |
| NEXT_TYPE | NOT NULL | CHAR (10) |
| TASK_BACK | NOT NULL | NUMBER (7) |

ANNEXE 8(suite)

5

SQL> desc sched_dyn_status;

10

| Name | Null? | Type |
|----------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| QUEUE | NOT NULL | NUMBER (3) |
| INDEX_ID | NOT NULL | NUMBER (8) |

15

SQL> desc sched_dyn_state;

20

| Name | Null? | Type |
|----------------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| AID | NOT NULL | NUMBER (8) |
| STATE | NOT NULL | CHAR (10) |
| TOKENS_CURRENT | NOT NULL | NUMBER (3) |
| TOKENS_PAST | NOT NULL | NUMBER(3) |

30

SQL> desc disp_circulation;

35

| Name | Null? | Type |
|------------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| AID | NOT NULL | NUMBER (8) |
| PREV_USRID | NOT NULL | NUMBER (7) |
| NEXT_USRID | NOT NULL | NUMBER (7) |
| OUT_BRANCH | NOT NULL | NUMBER (3) |
| IN_BRANCH | NOT NULL | NUMBER (3) |
| NEXT_TYPE | NOT NULL | CHAR (10) |
| INDEX_ID | NOT NULL | NUMBER (8) |
| TASK_BACK | NOT NULL | NUMBER (7) |

45

SQL> desc disp_dyn_circ;

50

| Name | Null? | Type |
|------------|----------|------------|
| PID | NOT NULL | NUMBER (5) |
| JID | NOT NULL | NUMBER (6) |
| PREV_USRID | NOT NULL | NUMBER (7) |
| NEXT_USRID | NOT NULL | NUMBER (7) |
| OUT_BRANCH | NOT NULL | NUMBER (3) |
| IN_BRANCH | NOT NULL | NUMBER (3) |
| NEXT_TYPE | NOT NULL | CHAR (10) |
| TASK_BACK | NOT NULL | NUMBER (7) |
| INDEX_ID | NOT NULL | NUMBER (8) |

55

ANNEXE 8(suite)

5

SQL> desc disp_dyn_statusc;

| | Name | Null? | Type |
|----|----------|----------|------------|
| | PID | NOT NULL | NUMBER (5) |
| | JID | NOT NULL | NUMBER (6) |
| 15 | QUEUE | NOT NULL | NUMBER (3) |
| | INDEX_ID | NOT NULL | NUMBER (8) |

SQL> desc disp_dyn_statec

| | Name | Null? | Type |
|----|----------------|----------|------------|
| | PID | NOT NULL | NUMBER (5) |
| 25 | JID | NOT NULL | NUMBER (6) |
| | AID | NOT NULL | NUMBER (8) |
| | USRID | NOT NULL | NUMBER (8) |
| | STATE | NOT NULL | CHAR (10) |
| | TOKENS_CURRENT | NOT NULL | NUMBER (3) |
| 30 | TOKENS_PAST | NOT NULL | NUMBER(3) |

Revendications

35

1. Système de traitement transactionnel entre un serveur (3) informatique et une pluralité de stations de travail (2) entre lesquelles les ouvrages correspondants à des activités constituant des procédures sont exécutés, le système comportant du côté serveur un module ordonnanceur des ouvrages qui s'appuient sur la définition des procédures (à savoir les précédences entre les activités d'une procédure) mémorisées dans la base de données relationnelle pour ordonner le progrès des ouvrages en trouvant la ou les activités suivantes et en créant la ou les tâches correspondantes étant donnée la fin d'une activité, un module répartiteur (31) pour effectuer la répartition des tâches qui s'appuient sur la relation entre rôle et acteur mémorisée dans la base de données relationnelle pour attribuer une tâche à un acteur choisi éventuellement selon la charge de travail, un module signaleur (34) pour conserver une trace des événements rythmés et recevant des informations de l'ordonnanceur (30) et d'un module coordinateur (31), une boîte de messages (35) échangeant des signaux d'une part avec les modules répartiteur (33) et signaleur (34), et d'autre part avec un module de communication (36) avec les stations de travail et un module de gestion des données du flux de travail (370), le système comportant du côté station de travail une couche de communication (27), une couche de gestion des messages (26) correspondant au flux de travail, une couche (24) comportant différents jeux d'interfaces de programmation de l'application, une couche de services de gestion (25) et une couche d'interface utilisateur (20,21).

40

2. Système selon la revendication 1 caractérisé en ce que les différents modules d'une couche de la station de travail (2) n'ont pas d'interaction entre eux.

45

3. Système selon la revendication 1 caractérisé en ce que le module de gestion des données (370) gère des tables statiques et dynamiques mémorisées dans la base de données relationnelle (372).

50

4. Système selon la revendication 3 caractérisé en ce que les tables statiques sont constituées des données

d définition des procédures, d définition des activités, d l'organisation des acteurs et des rôles et de répartition des activités selon un tableau de distribution.

5. Système selon la revendication 3 caractérisé en ce que les tables dynamiques sont constituées de données évolutives représentant l'avancement des ouvrages et des tâches.

10. Système de traitement transactionnel selon la revendication 3 caractérisé en ce que les tables dynamiques sont constituées d'une table de circulation, d'une table de vecteur de données d'ouvrage JDV-JDV, d'une table de vecteur de données d'activité ADV-ADV et des tables SCHED-STATEP, SCHED-STATUSP, SCHED-JOBS, SCHED-TASKP.

15. Système de traitement transactionnel selon la revendication 4 caractérisé en ce que les tables statiques sont constituées d'une table de description de procédure SCHED-PROCEDURE, de description de données de procédure JDVPDDEF, d'une table de description des références des données stockées dans une base de données externe (EDA-ORA), d'une table de description des activités SCHED-ACTIVITE, d'une table de définition des précédences SCHED-PRECEDENCE, et des tables de définition de l'organisation du système ORGA-ACTOR, ORGA-ROLE, ORGA-R-OFF-ACT.

20. Système de traitement transactionnel selon la revendication 4 caractérisé en ce que une activité est constituée d'un identifiant de l'activité et de la procédure à laquelle l'activité est rattachée et d'une liste d'actions comportant chacune une commande d'initialisation et une commande de validation.

25. Système de traitement transactionnel selon la revendication 8 caractérisé en ce que les actions sont constituées du type, d'un nom, des arguments constitués par les données du vecteur JDV ou ADV.

30. 10. Système de traitement transactionnel selon la revendication 1 caractérisé en ce que la couche de service de gestion comporte :

- un service de gestion de création d'ouvrage,
- un service de gestion d'administration ,
- un service de gestion de coordination,
- un service de gestion de l'exécution de l'activité.

35. 11. Système de traitement transactionnel selon la revendication 8 caractérisé en ce que une activité comporte également un préfixe (PR) destiné à appeler un logiciel de présentation et d'interaction fournissant l'interface de l'activité.

40. 12. Système de traitement transactionnel selon la revendication 1 caractérisé en ce que le module administrateur réalise les fonctions de modelage, de création, test et accord des différentes entités du système.

45. 13. Système de traitement transactionnel selon la revendication 1 ou 12 caractérisé en ce que le module de gestion des données a pour fonction d'attacher les données appropriées aux ouvrages spécifiques en créant le vecteur JDV, de gérer l'intégrité des données, de retrouver et accéder les données dans les bases de données de l'utilisateur.

50. 14. Système de traitement transactionnel selon la revendication 1 ou 13 caractérisé en ce que le module de gestion d'exécution d'activité comporte les fonctions

- de lecture du fichier de description d'activité,
- de lecture et d'écriture des données d'ouvrage utilisé dans l'activité,
- de lecture et d'écriture des données des tâches correspondantes utilisées dans l'activité et de commande des droits d'accès sur les données d'ouvrage,
- de lancement du formulaire de présentation appropriée à l'activité et de l'exécution des actions correspondantes à l'activité d'une façon guidée.

55. 15. Système de traitement transactionnel selon la revendication 1 ou 14 caractérisé en ce que le module de gestion de tâches exécute les fonctions suivantes :

- réception d'une tâche offerte,
- acceptation d'une tâche offerte ,
- recherche des données correspondantes du vecteur d'ouvrage sur le serveur,
- achèvement d'une tâche,

5 - restitution du vecteur d'ouvrage mis à jour au serveur,
 - réorientation d'une tâche vers un autre acteur ou une autre activité,
 - appel de l'activité d'exécution d'une tâche correspondante par l'interface utilisateur,
 - traitement et maintient local des données correspondant aux tâches et leur historique .

10 16. Système de traitement transactionnel selon la revendication 1 ou 15 caractérisé en ce que le module administrateur et coordinateur comporte les fonctions d'édition de concept et de données et la commande des ouvrages.

15 17. Système de traitement transactionnel selon la revendication 1 ou 16 caractérisé en ce que le module répartiteur réalise trois fonctions principales :
 - affecter à un utilisateur les tâches qui viennent d'être créer par l'ordonnanceur,
 - affecter un nouvel utilisateur à une tâche qui vient d'être rejetée,
 - effectuer la distribution et la circulation des activités parmi les utilisateurs.

20 18. Système de traitement transactionnel selon la revendication 1 ou 17 caractérisé en ce que le module signalisateur conserve une trace des événements rythmés et comporte une fonction de rappel quand un temps est expiré et une fonction de retardement pour lancer un travail avec un retard.

25 19. Système de traitement transactionnel selon la revendication 1 ou 19 caractérisé en ce que la boîte de message est constituée de deux tables de données, une première qui contient la description des messages et une seconde qui mémorise les destinataires des messages.

25

30

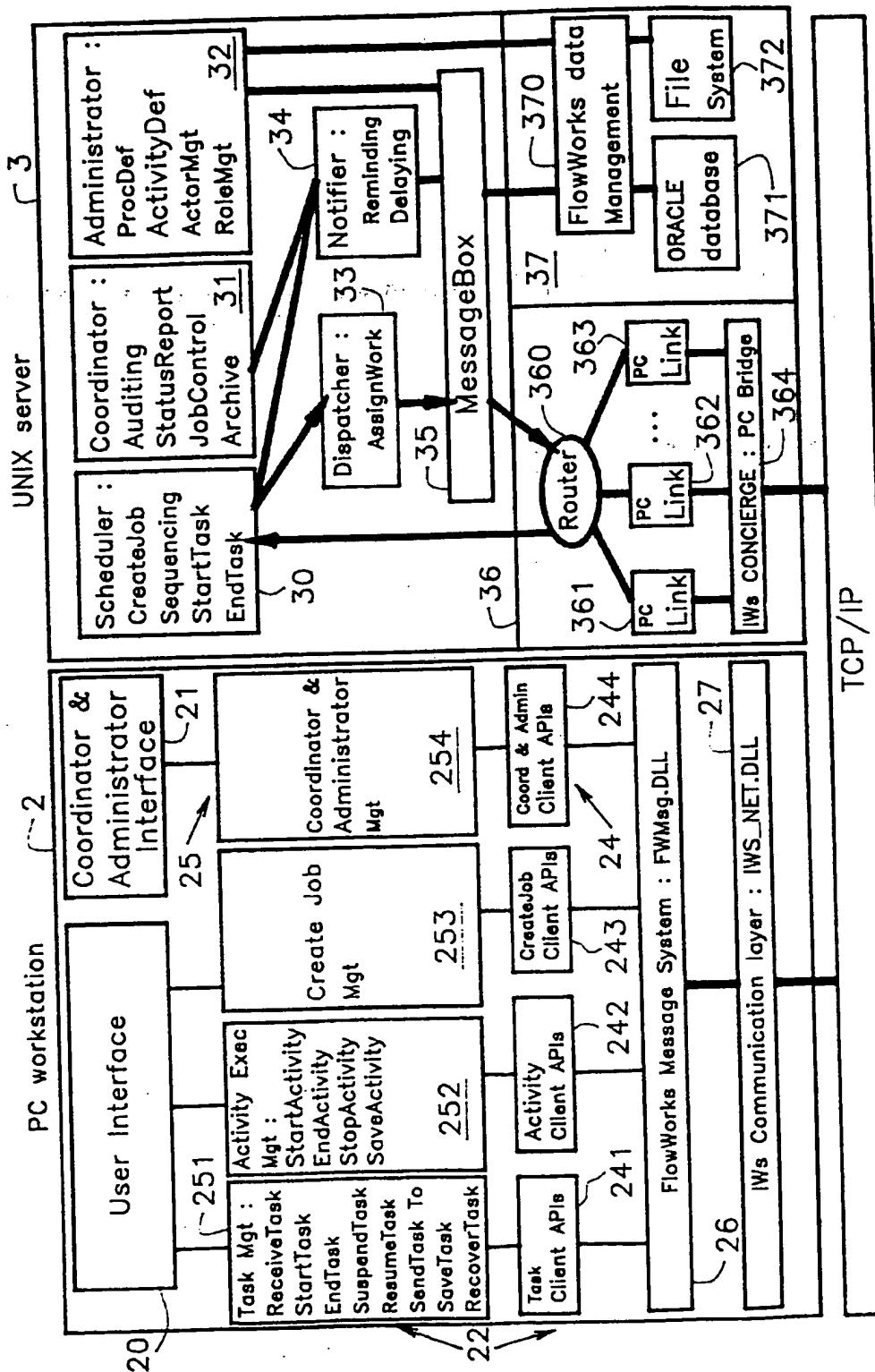
35

40

1

50

55



一
EIG
E

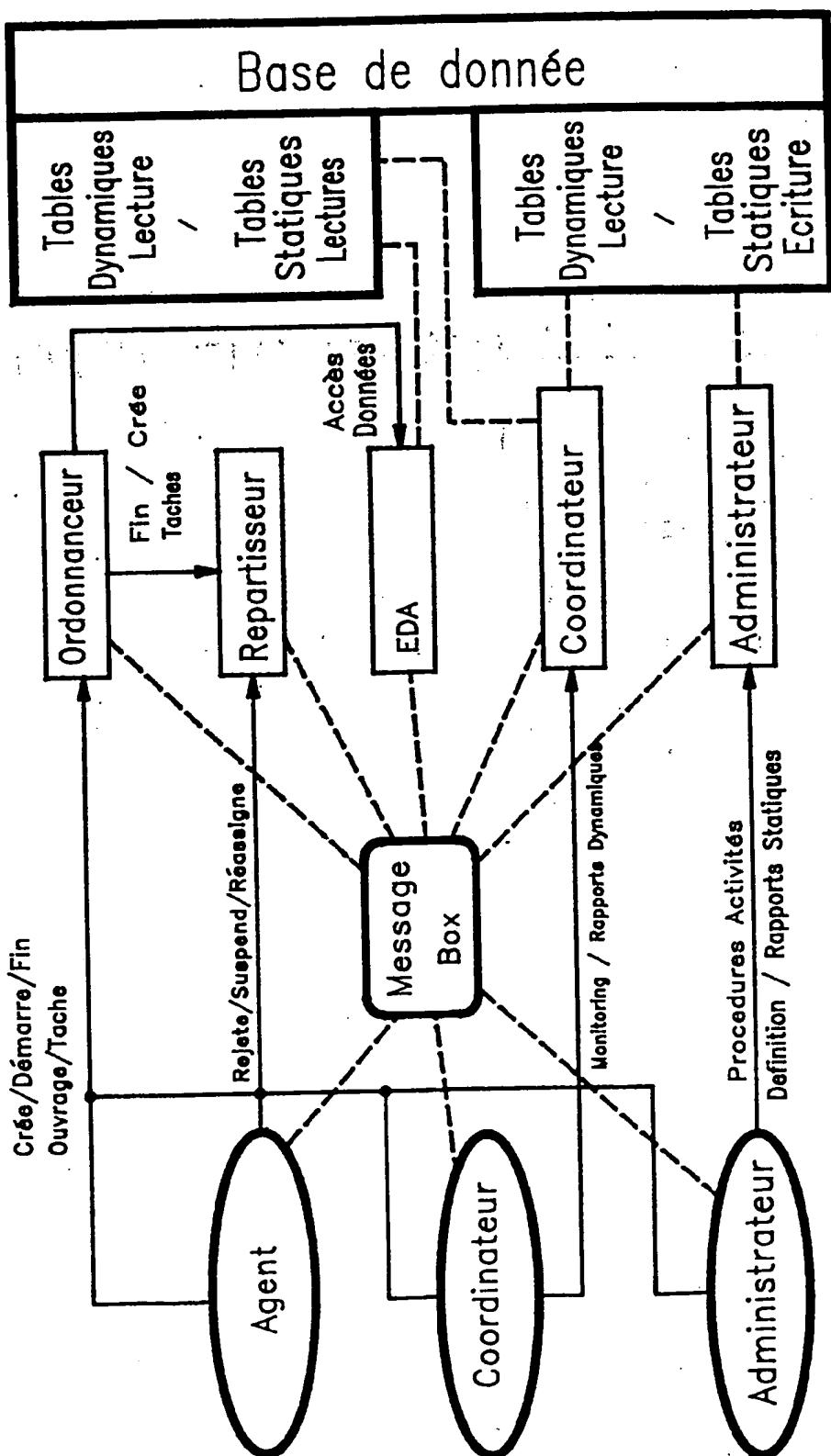


FIG.2

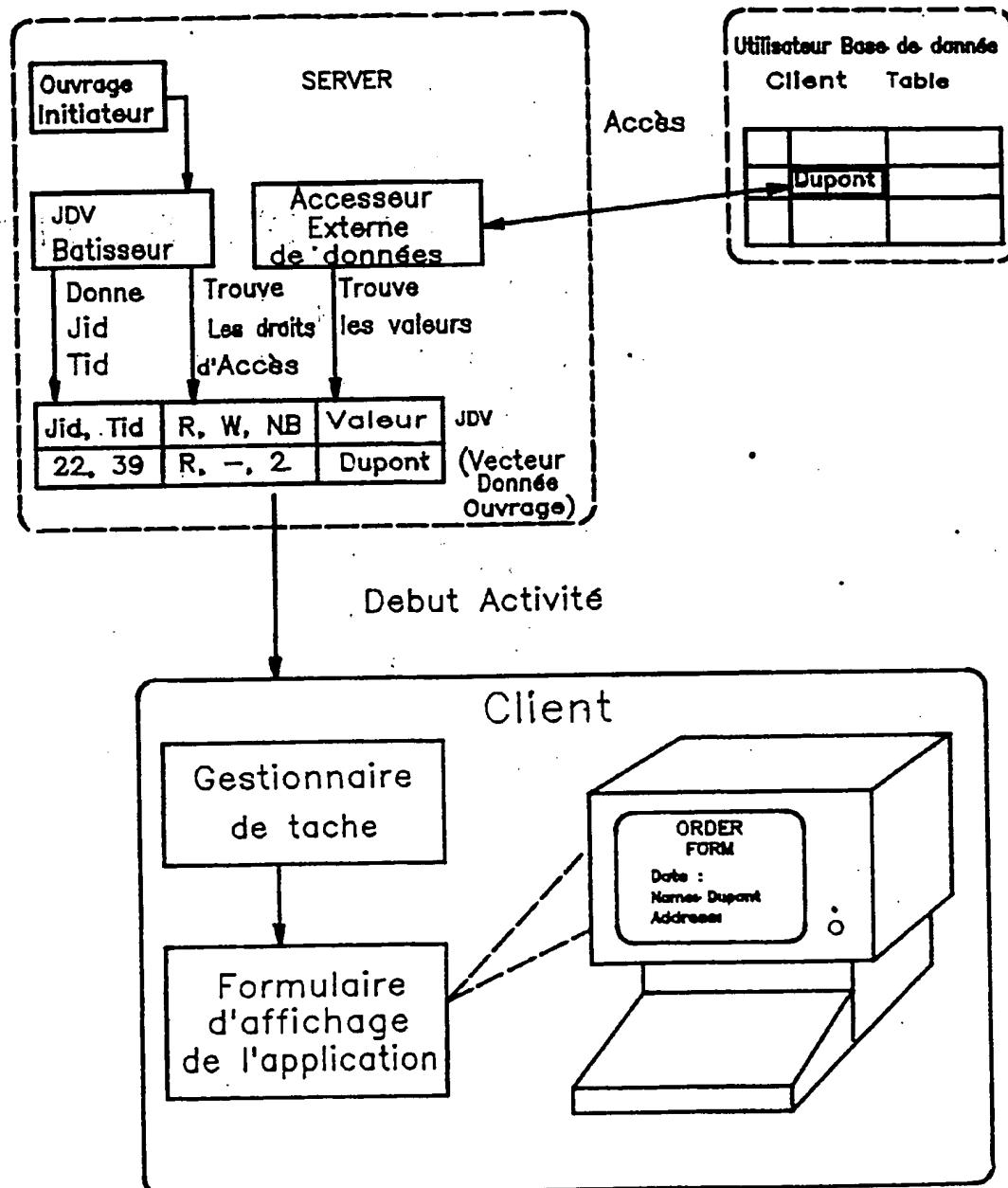


FIG.3

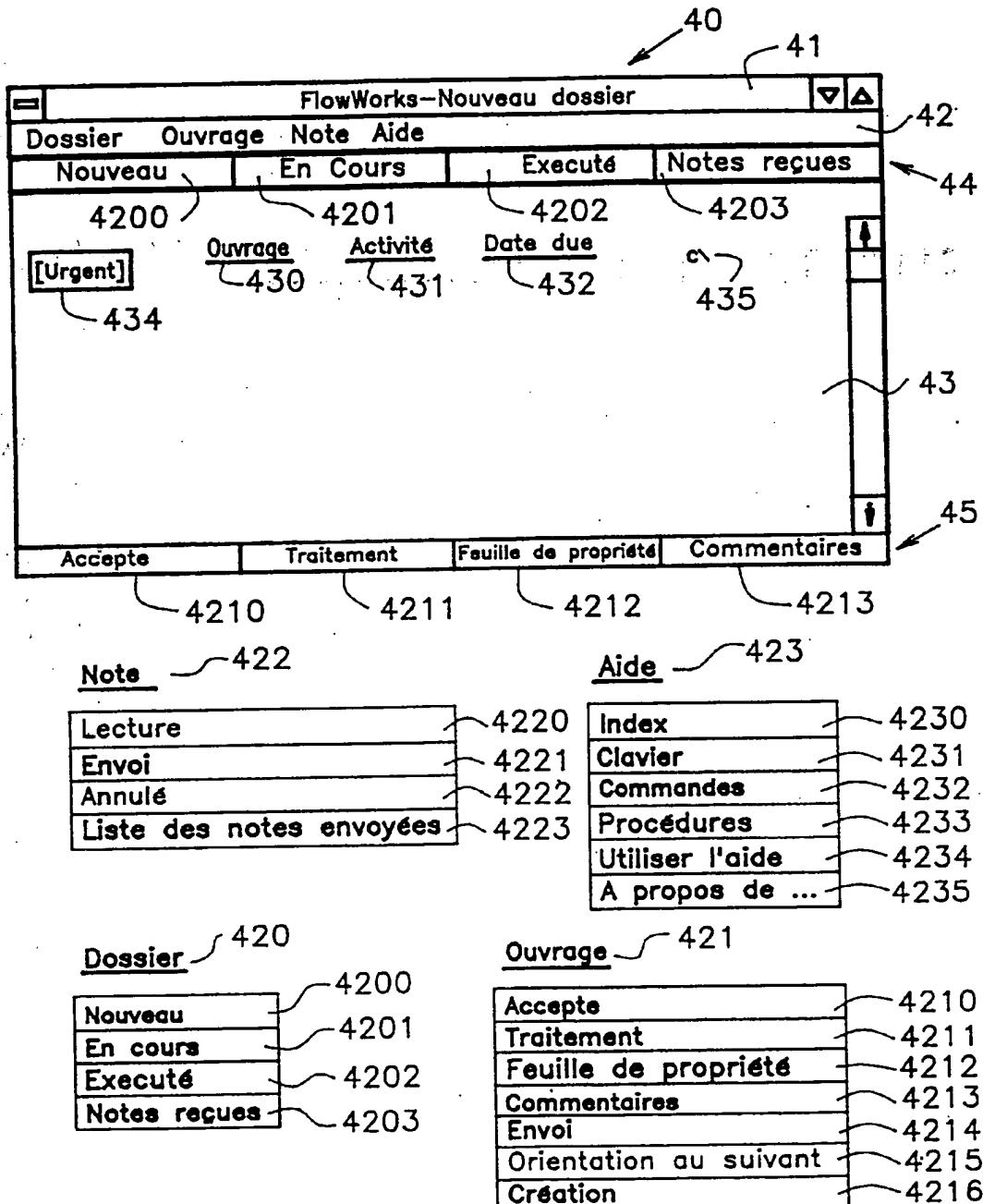


FIG.4

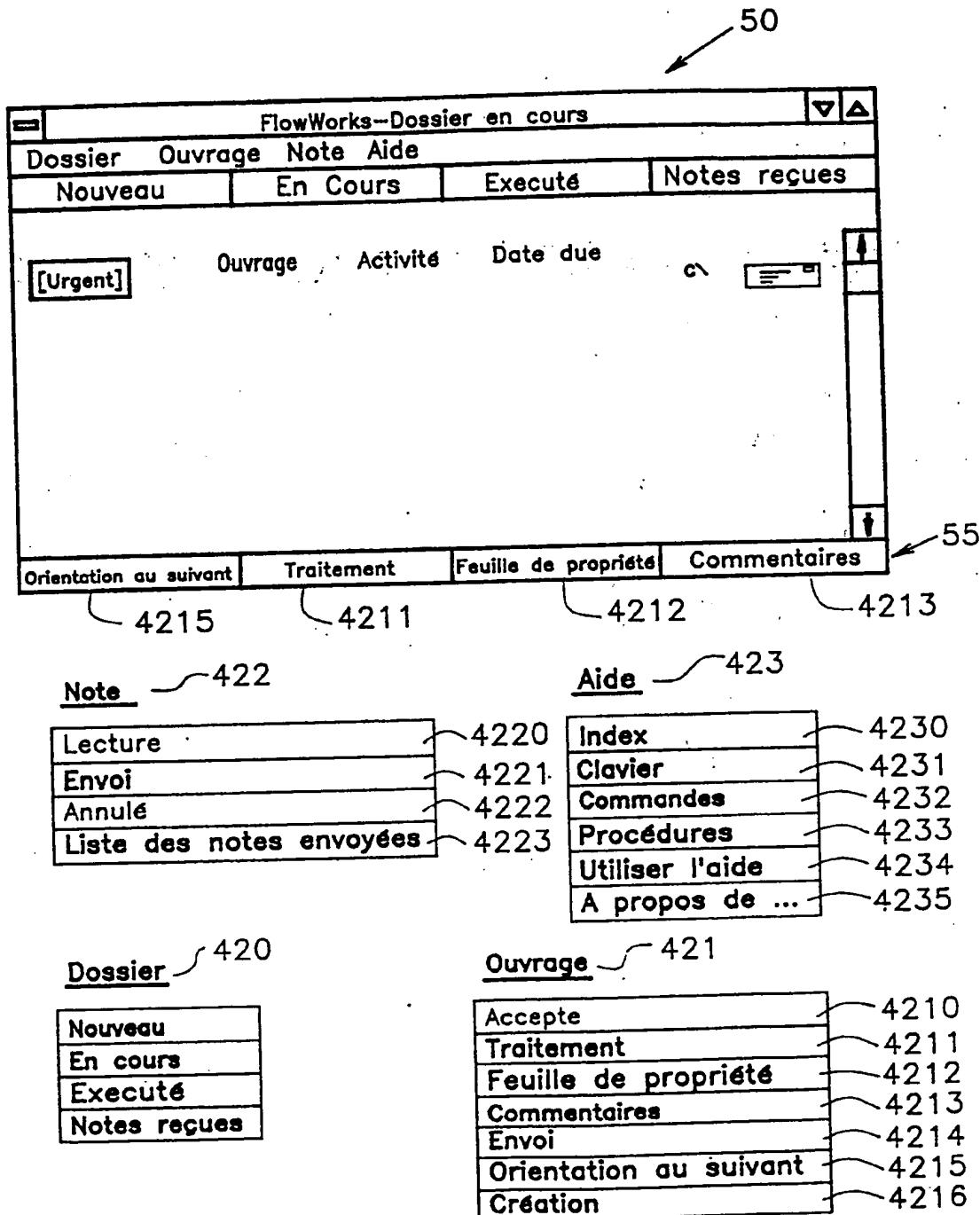


FIG.5

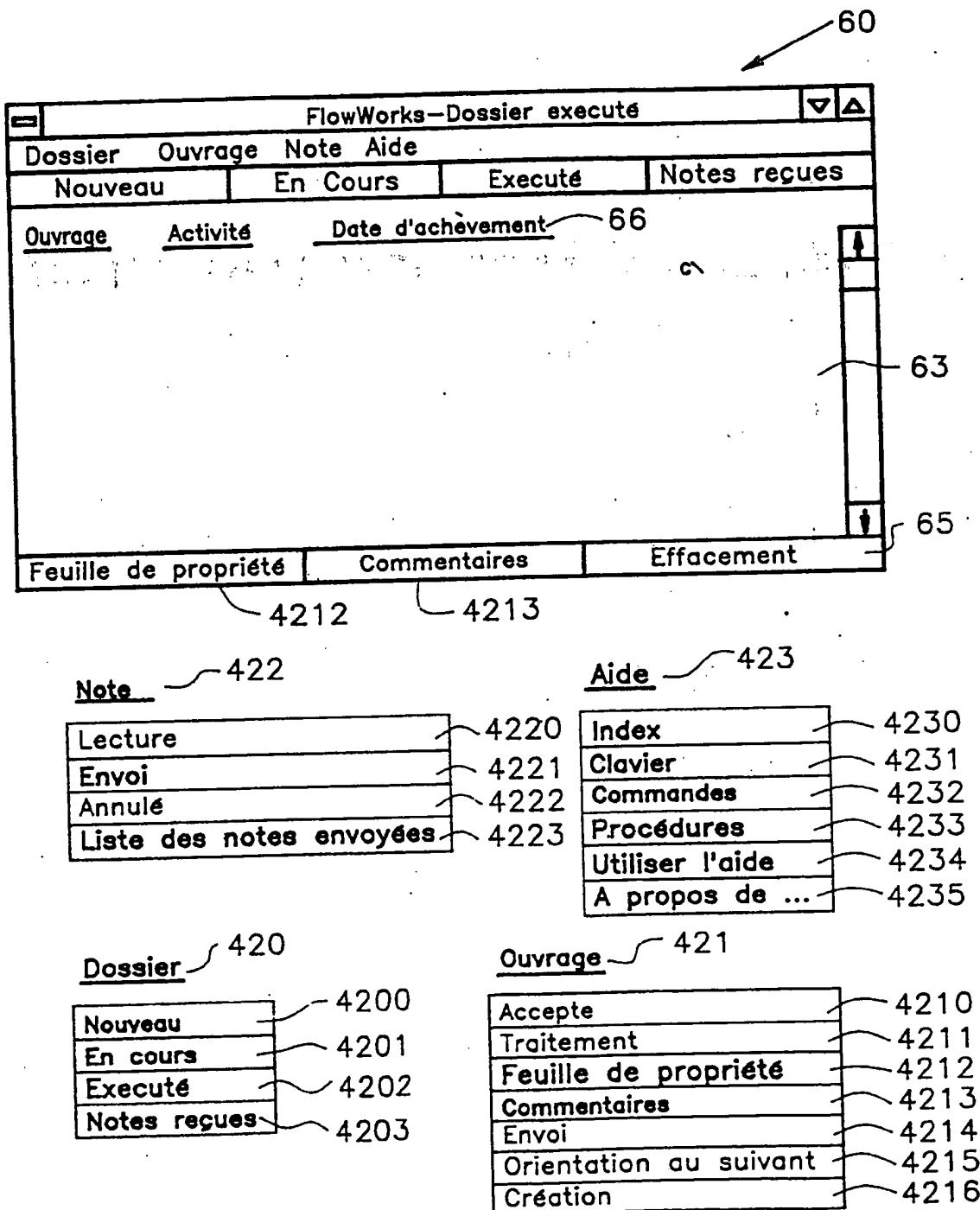


FIG.6

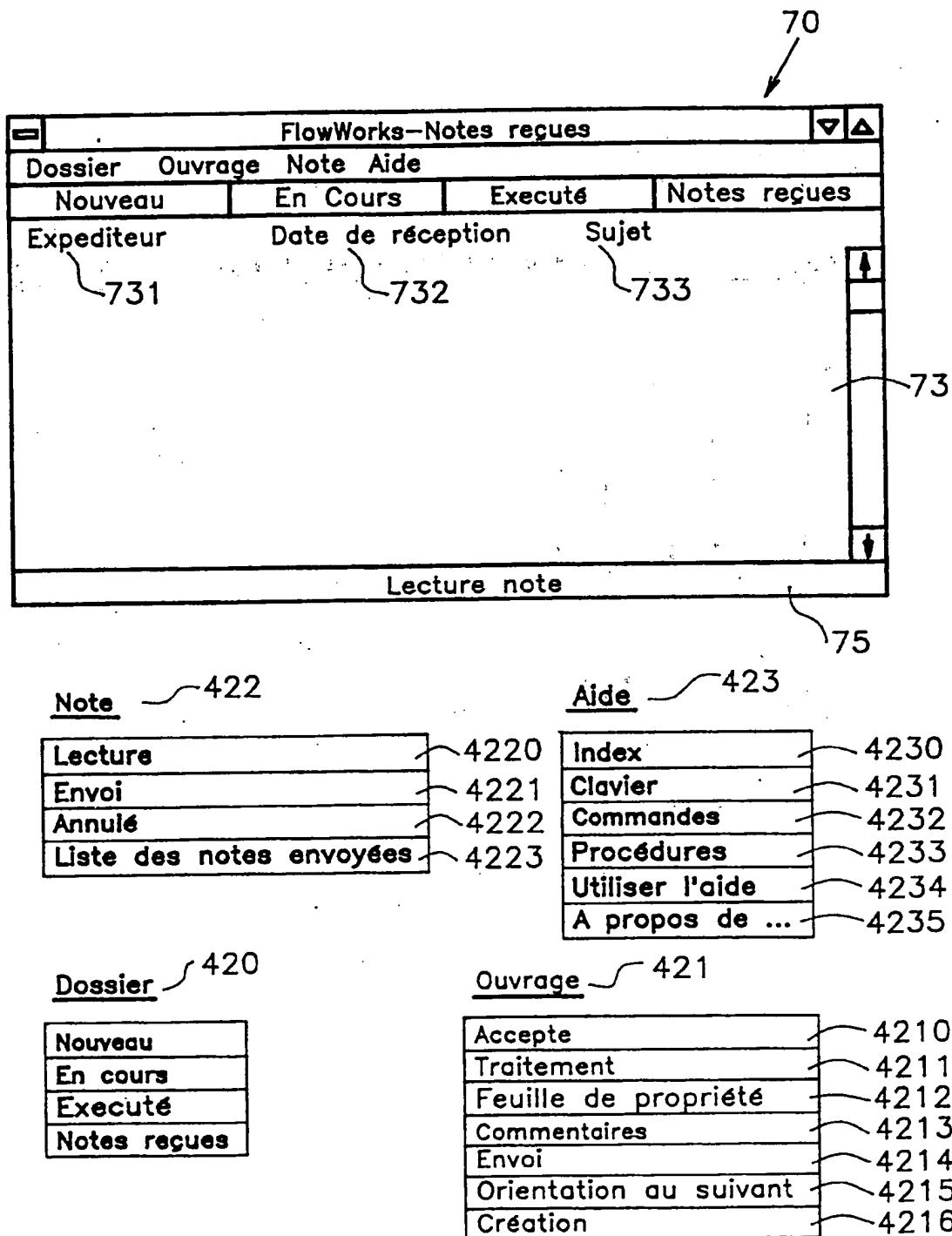
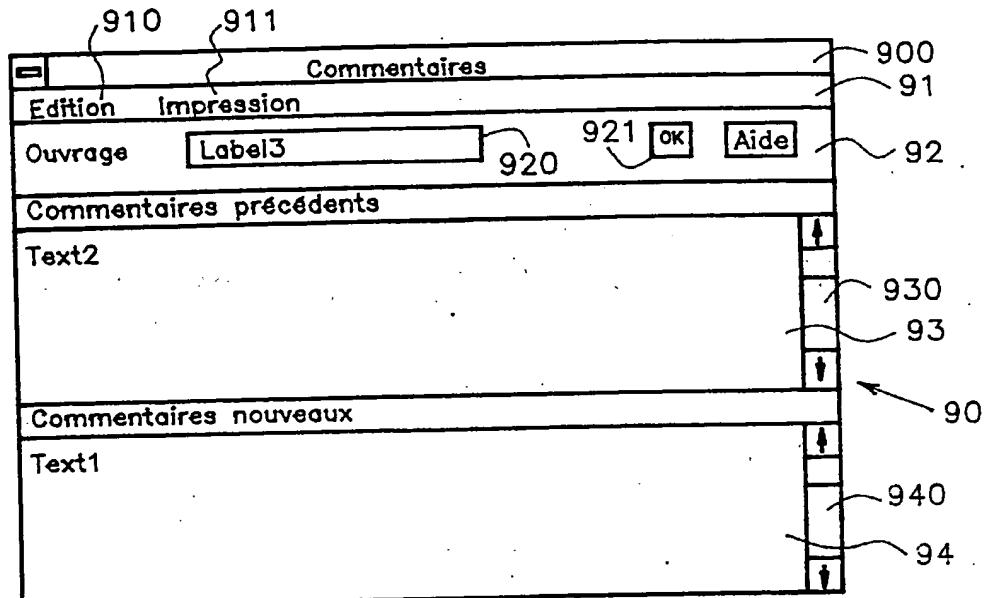


FIG.7



Edition 910 Impression 911

| | | | |
|---------|------|-------------------------|------|
| Couper | 9101 | Commentaires précédents | 9111 |
| Copier | 9102 | Nouveaux commentaires | 9112 |
| Coller | 9103 | Sélection de tous | 9113 |
| Effacer | 9104 | | |

FIG.9

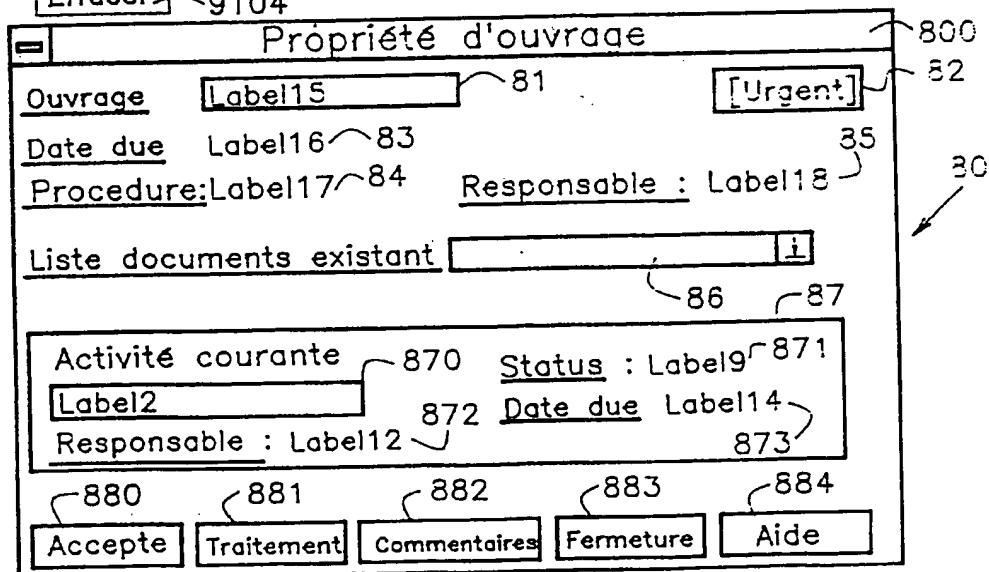


FIG.8

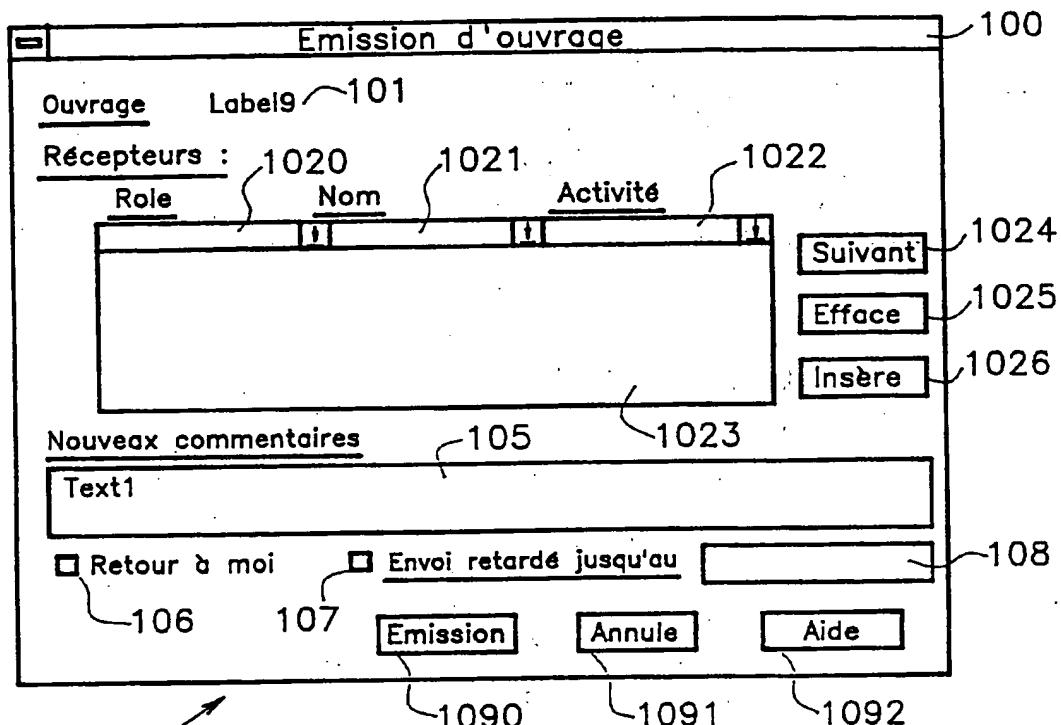


FIG. 10

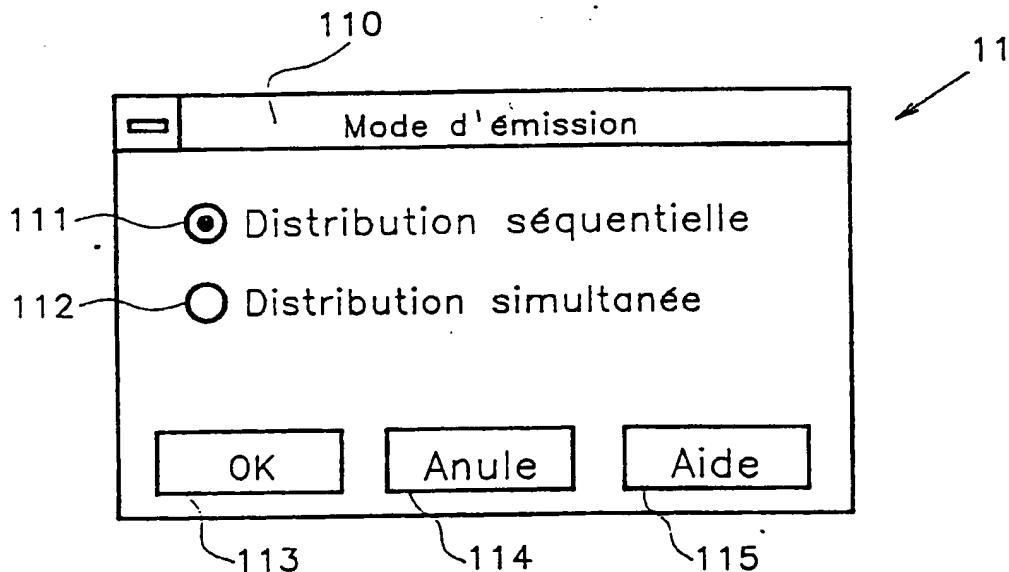


FIG. 11

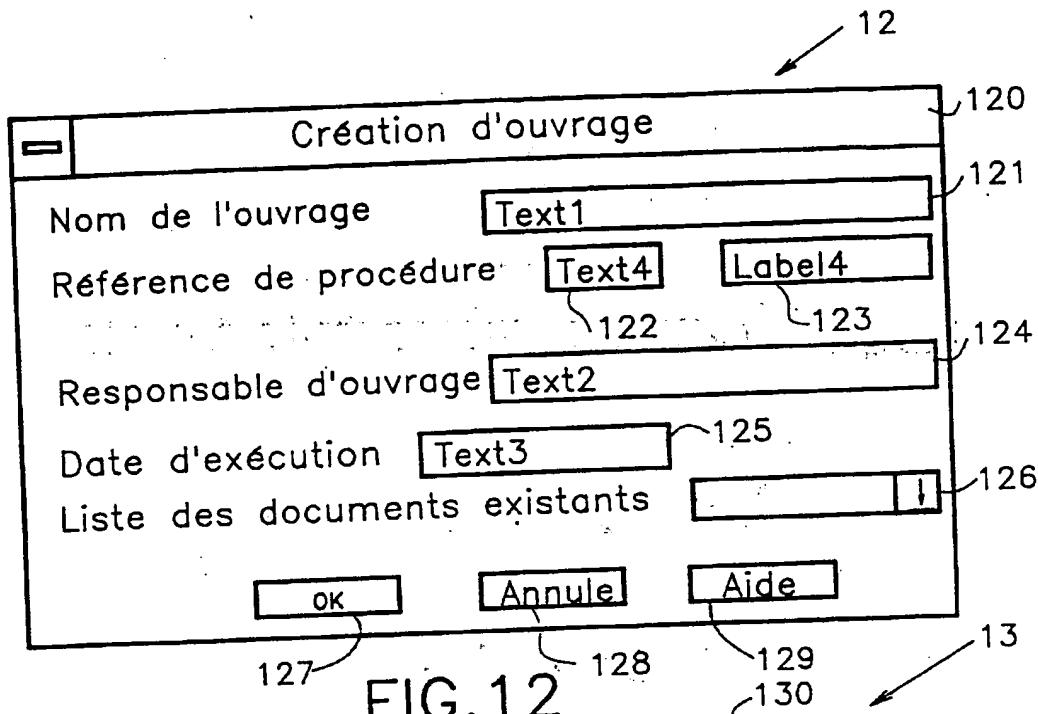


FIG. 12

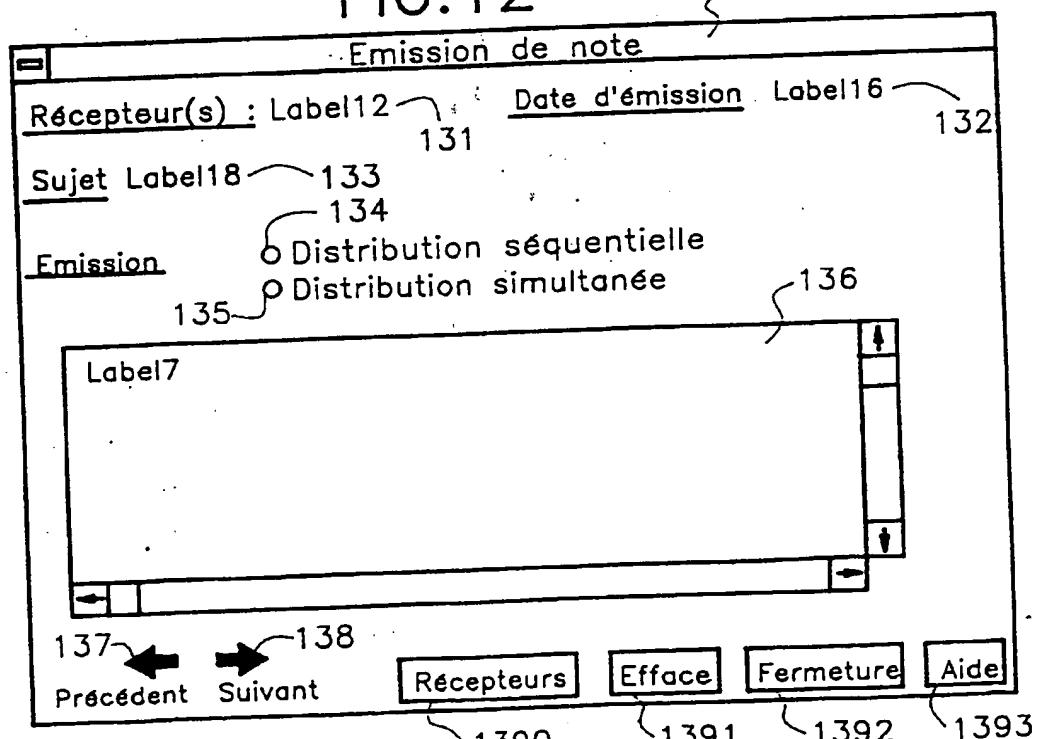


FIG. 13

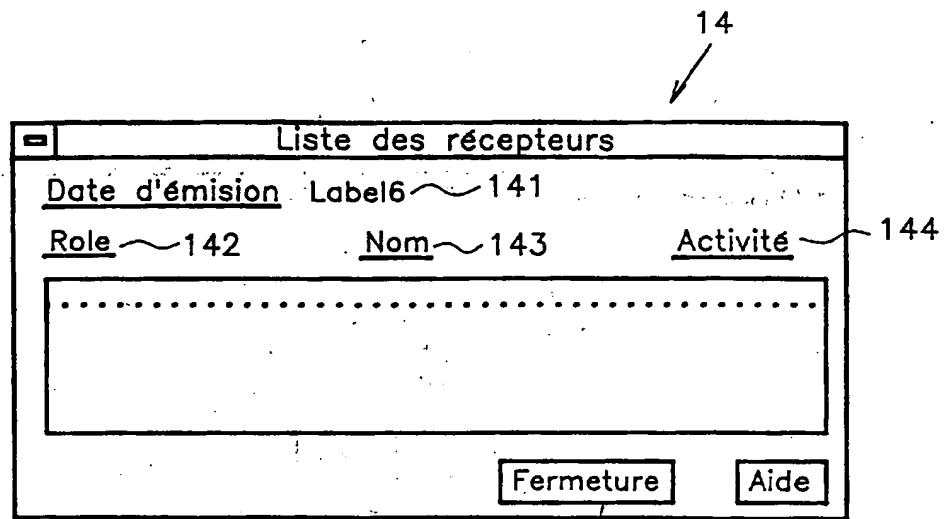


FIG.14 145

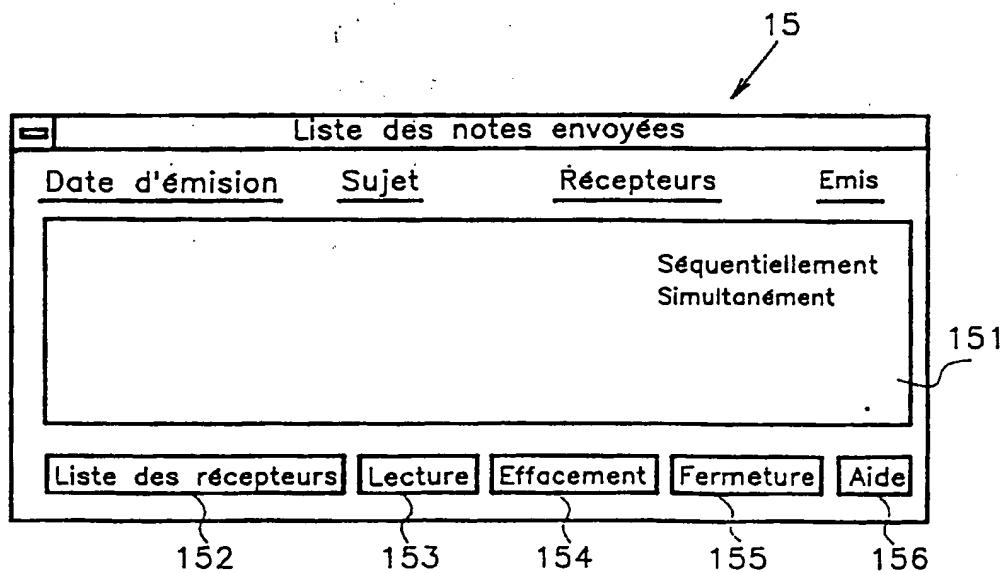


FIG.15



Office européen
des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numéro de la demande

EP 93 40 1459

| DOCUMENTS CONSIDERES COMME PERTINENTS | | | CLASSEMENT DE LA DEMANDE (Int. Cl.5) | | | |
|--|---|-----------------------------|--|--|---|-----------------------------|
| Catégorie | Citation du document avec indication, en cas de besoin, des parties pertinentes | Revendication concernée | | | | |
| A | EP-A-0 178 235 (INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE) 16 Avril 1986 * page 1, ligne 28 - page 4, ligne 3 * | 1 | G06F9/46 | | | |
| A | WO-A-9 207 324 (TEKNEKRON SOFTWARE SYSTEMS, INC.) 30 Avril 1992 * abrégé; figure 17 * | 1 | | | | |
| | | | DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5) | | | |
| | | | G06F | | | |
| <p>Le présent rapport a été établi pour toutes les revendications</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Lieu de la recherche LA HAYE</td> <td style="width: 33%;">Date d'achèvement de la recherche 13 JUILLET 1993</td> <td style="width: 33%;">Examinateur WILTINK J.G.</td> </tr> </table> | | | | Lieu de la recherche LA HAYE | Date d'achèvement de la recherche 13 JUILLET 1993 | Examinateur WILTINK J.G. |
| Lieu de la recherche LA HAYE | Date d'achèvement de la recherche 13 JUILLET 1993 | Examinateur WILTINK J.G. | | | | |
| <p>CATEGORIE DES DOCUMENTS CITES</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrêté-plan technologique O : divulgation non-écrite P : document intercalaire </td> <td style="width: 50%; vertical-align: top;"> T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant </td> </tr> </table> | | | | X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrêté-plan technologique O : divulgation non-écrite P : document intercalaire | T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant | |
| X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrêté-plan technologique O : divulgation non-écrite P : document intercalaire | T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant | | | | | |